NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A DATA WAREHOUSING AND OLAP APPLICATION FOR THE NAVAL RESERVE FORCE (CNRF) ASSESSMENT PROCESS

by

Scott A. Langley

September 2000

Thesis Advisor:	Dan Dolk
Associate Advisor:	Magdi Kamel

Approved for public release; distribution is unlimited

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
30 Sep 2001	N/A	-

Title and Subtitle A Data Warehousing And Olap Application For The Naval Reserve Force (CNRF) Assessment Process	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Langley, Scott A.	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Research Office Naval Postgraduate School Monterey, Ca 93943-5138	**Performing Organization Report Number**
Sponsoring/Monitoring Agency Name(s) and Address(es)	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)
Distribution/Availability Statement Approved for public release, distribution unlimited	
Supplementary Notes	
Abstract	
Subject Terms	
Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU
Number of Pages 139	

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2001	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: A Data Warehousing And Olap Application For The Naval Reserve Force (CNRF) Assessment Process	**5. FUNDING NUMBERS**
6. AUTHOR(S) Langley, Scott A.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Commander, Naval Reserve Force 4400 Dauphin Street New Orleans, LA	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

The Naval Reserve Force has identified a need to pool the data from its many legacy database systems into a single, useable data warehouse. The current system uses separate legacy databases and formatted reports to provide a manual decision process. Under the leadership of Rear Admiral John Totushek, the Naval Reserve Force is driving many technological revolutions via the Leading Change initiative. One of the key goals of the Leading Change initiative is a strategic decision support tool. To support this goal, Naval Reserve Force Assessment Division elected to fund a project to provide a prototype data warehouse and Online Analytical Processing (OLAP) solution to the problem. The Naval Reserve Strategic Decision Support Tool (NaRSDAT) is the result. The NaRSDAT development of this thesis provides an in depth evaluation of the existing databases. It then provides an object oriented development approach to a relational data warehouse and a star schema development for data mining. NaRSDAT employs Microsoft Visual Basic, Microsoft Access, and Cognos PowerPlay to provide a complete data warehouse and OLAP solution. The NaRSDAT prototype will serve as the basis for a comprehensive knowledge management solution for the Naval Reserve Force.

14. SUBJECT TERMS Data warehousing, On-Line analytical processing (OLAP)	**15. NUMBER OF PAGES** 121
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A DATA WAREHOUSING AND OLAP APPLICATION FOR THE NAVAL RESERVE FORCE (CNRF) ASSESSMENT PROCESS

Scott A. Langley
Lieutenant Commander, United States Naval Reserve
B.S., United States Naval Academy, 1986

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2001

Author: _____
Scott A. Langley

Approved by: _____
Dan Dolk, Thesis Advisor

_____
Magdi Kamel, Associate Advisor

_____
Dan Boger, Chairman
Information Systems Academic Group

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Naval Reserve Force has identified a need to pool the data from its many legacy database systems into a single, useable data warehouse. The current system uses separate legacy databases and formatted reports to provide a manual decision process. Under the leadership of Rear Admiral John Totushek, the Naval Reserve Force is driving many technological revolutions via the Leading Change initiative. One of the key goals of the Leading Change initiative is a strategic decision support tool. To support this goal, Naval Reserve Force Assessment Division elected to fund a project to provide a prototype data warehouse and Online Analytical Processing (OLAP) solution to the problem. The Naval Reserve Strategic Decision Support Tool (NaRSDAT) is the result. The NaRSDAT development of this thesis provides an in depth evaluation of the existing databases. It then provides an object oriented development approach to a relational data warehouse and a star schema development for data mining. NaRSDAT employs Microsoft Visual Basic, Microsoft Access, and Cognos PowerPlay to provide a complete data warehouse and OLAP solution. The NaRSDAT prototype will serve as the basis for a comprehensive knowledge management solution for the Naval Reserve Force.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The author would like to recognize Rear Admiral John Totushek for his insightful leadership. His Leading Change initiatives provided the necessary impetus to make this a successful project and build the foundation for a force wide strategic decision support system.

The author would also like to thank the staff of CNRF N8 for their assistance with data acquisition and database resolution. A special thanks goes to Capt John W. Coleman, II, N8 DCOS, for his assistance with funding and technical support. His mentorship in this endeavor made the process a true learning experience.

Thanks to Dr. Donn Peery for his assistance in creating this document and moral support during the academic process. Lastly, I would like to thank Sally, Dennis, Stuart, Moesha, Frank, and Al who tolerated my absence of body and mind during the hours of work.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

As a Navy component, the Naval Reserve Force uses both general Navy information systems and independent Naval Reserve information systems. Traditionally, these information systems are stand-alone transaction database systems. They serve one primary purpose and are isolated from all other Navy or Naval Reserve information systems.

Commander, Naval Reserve Force (CNRF) Assessment Division (N8) is charged with assessing the performance of the Naval Reserve. The assessment process involves gathering data from 6 existing databases in remote locations, filtering critical data from each database, and presenting the data in a format that can be interpreted by the users. The current assessment process takes three people several weeks to complete.

The assessment process is a key factor in the Naval Reserve's strategic planning process. The assessment produced by N8 allows the planning team to rate their performance over the past year and determine what changes should be made to provide a better service to their customers, both internal and external. They also look at trends from previous years and predict action or inaction in key performance areas to stabilize the Naval Reserve Force performance.

The assessment provides such key statistical information as dollars spent per Selected Reservist, dollars spent per enlisted rating, East coast/West coast cost comparisons for Active Duty Training (ADT), and percent billet fills per rating. This information is then used to determine budgetary requirements for annual training, rate recruiting targets, Naval Reserve reorganization requirements, and many other strategic operating considerations. This information can also be used to display a failure in policy or a misunderstanding of the published policy by the commands in the field.

N8 has recently identified a requirement to automate their assessment process. The requirement is based on the need for a more standardized product from the assessment process as well as a more flexible, accurate, and timely decision-making tool

at the Headquarters level. Current military manning levels and high personnel turnover rates also played a part in the identification of this requirement.

The current process is lengthy and unresponsive. A full assessment is only performed once per year. Problem areas are targeted for intermediate updates as time and manpower becomes available throughout the year. Since the year-end process takes several months, it cannot be used to set the current year policy for operations. By the time the assessment is available for evaluation, it is too late to affect the current year operations. The lessons from that assessment are then applied to the next operational year. During the interim period, many of the internal and external environmental factors that caused the assessment results have changed and may not lead to a proper application of the assessment results. Naval Reserve Force staff is continually playing a game of catch-up and can become confused concerning the effectiveness of the changes made and the reliability of the assessment information. This leads to a poor strategic decision making environment.

B. OBJECTIVES

The purpose of this thesis is to provide CNRF N8 with a working software prototype that will reduce the time and manpower necessary for producing the annual Naval Reserve assessment, while concurrently increasing the quality and timeliness of the information that is generated by the assessment. The Naval Reserve Strategic Decision Assistance Tool (NaRSDAT) will provide a data warehouse and On-Line Analytical Processing (OLAP) approach to satisfy these objectives. NaRSDAT will also provide a tool that can be used on a more frequent basis to assist in the strategic decision making process. The current period requirement for assessment updates is quarterly. NaRSDAT will allow updates as frequently as the legacy database data is uploaded. It will provide the decision maker with the ability to incorporate current, relevant information in the decision making process in order to streamline operations, reduce risk, and provide both internal and external customers with the best available product.

The NaRSDAT application answers the following three research questions:

- What design methodology and performance metrics are appropriate for the warehousing and OLAP process?
- What data migration strategy is appropriate for transferring elements from the existing databases to the data warehouse in order to meet the assessment requirements?
- What data quality standards must be imposed to maintain the data integrity of the warehouse?

C. SCOPE

This thesis describes a stand-alone application prototype called NaRSDAT, which is a tool to assist CNRF N8 in accomplishing the assessment mission. NaRSDAT automates the data cleansing, relational database import function, OLAP data cube development, and common report and query generation. NaRSDAT components include a Visual Basic code section for cleaning and importing data, a Microsoft Access™ database for data warehousing, a Cognos Powerplay ™ data cube for OLAP, and queries in both Microsoft Access ™ and Cognos Powerplay ™ for report generation.

NaRSDAT will not automate the data retrieval from the legacy databases. That requirement is beyond the current scope of work. The existing manual process of data acquisition will continue to be used. This thesis also does not provide a distributed decision support solution. The stand-alone system will have the ability to be expanded at a later date through common, industry standard practices to include a distributed network solution as well as an Internet based solution. However, the development of such a system is beyond the scope of work for this thesis.

D. METHODOLOGY

A search of modern literature for OLAP, data warehousing, data mining, and data quality will be conducted. All available information on the six legacy databases will also be acquired. The specific requirements of NaRSDAT will be compared to generic case studies from the literature and a specific development will be applied.

This thesis will use a standard object-oriented development approach to software design. Unified Modeling Language (UML) will be used as the core tool for the software design. Use cases will be developed to assist with defining the system functional requirements. A specific data model will then be developed to support the data warehouse. This database will be converted into a format that can be imported into the OLAP tool for data mining requirements.

All software will be developed in accordance with the Department of Defense's (DoD) Information Technology Standards Guidance [1]. All areas of ODBC, design, distribution, and compression will be compliant with these standards.

E. ORGANIZATION

This thesis is organized as follows:

Chapter II provides a review of the Naval Reserve Force mission and the initiatives that led to the motivation for this thesis. It also details information about the six legacy databases that are being used in NaRSDAT.

Chapter III develops the actor diagram and describes the primary actors in the NaRSDAT development.

Chapter IV develops the use cases for NaRSDAT. This section will describe the most common functions of the system and the actors that will participate in those functions.

Chapter V presents the functional requirements for the system. The functional requirements include internal system requirements as well as reports and charts required for the assessment process.

Chapter VI presents the data model for the relational database for the data warehouse. It will also show the star schema data model that will be imported into the OLAP product. Finally, this section will present the data cube model that will be used in the OLAP product.

Chapter VII generates data cleansing requirements for importing data from the existing databases. It discusses problems with missing and corrupt data as well as the reasoning for resolving and marking this type of data for future reference.

Chapter VIII describes how the prototype will be implemented at CNRF N8. It covers common use, limits of use, and user interfaces.

Chapter IX describes next generation developments at CNRF, distributed solution options, and future research areas.

Appendices provide a user's guide to the system, including installation instructions, operating instructions, Microsoft™ and Cognos™ help information, and troubleshooting assistance. Additional appendices provide legacy database metadata, Visual Basic code, and example reports.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. NAVAL RESERVE FORCE

The Naval Reserve Force's (NRF) primary mission is "to provide mission-capable units and individuals to the Navy, Marine Corps Team throughout the full range of operations from peace to war [2]." The Naval Reserve Force is divided into two primary forces: Naval Air Reserve Force and Naval Surface Reserve Force. Each of these forces is further subdivided into functional or type units. Each unit has it own mission requirements within the scope of the entire NRF mission.

B. LEADING CHANGE

Rear Admiral John B. Totushek assumed command of the Naval Reserve Force in October 1998. He is also Chief of Naval Reserve and Director, Naval Reserve. This three hatted position gives him complete developmental control over the Naval Reserve within Congressional budgetary and Department of the Navy policy boundaries. He is a firm believer in "Leading Change", an organizational change philosophy by John P. Kotter [3].

In keeping with this philosophy, RADM Totushek developed a vision statement [4] that directly supports the "Leading Change" philosophy. The vision statement is divided into three parts: the Naval Reserve mission, the Naval Reserve vision, and the "Leading Change" initiative. The mission is to provide trained troops throughout the range of operations. The vision defines a level of performance for the Naval Reserve that will provide the appropriate level of service to its customers. The last of the vision statements is "operating under leaders utilizing an overarching management system of technologically advanced business processes". Each of the requirements in the vision statement is supported by developing an arena of leadership in which knowledge, commitment to fulfilling customer requirements, continuous improvement, high quality, directed output, and monitoring and managing results are the driving principles.

C. THE INITIATIVE

The third part of the vision statement is the "Leading Change" initiative. This initiative further defines the direction of the Naval Reserve by providing a set of strategic concepts to guide the Naval Reserve. The Force Executive Steering Committee (ESC) directs the initiative. The ESC is made up of key leadership personnel from various areas of the Naval Reserve. The initiative sets a path for the establishment of a continuous improvement process. The path consists of the vision statement, a set of goals, a brief, a newsletter, and a feedback loop. The brief and the newsletter are designed to ensure that all hands can, and will, be involved in the change process. The feedback loop allows for input from any interested stakeholder.

Two of the goals outline the need for an assessment process. The assessment process must use available information from the Navy and Naval Reserve to properly gauge the status of the Naval Reserve and to record and demonstrate the success or failure of the various projects that support the "Leading Change" initiative. The Navy and the Naval Reserve use numerous databases to track performance and record key operational and budgetary information. A manpower intensive and time-consuming effort was established to develop cross-database knowledge about key performance metrics.

D. NRF DATABASES

The Naval Reserve uses both US Navy legacy database systems as well as internally generated legacy systems. Each system is isolated and used by the division or department to which the information directly relates. As with any other business, these databases have become "stove pipes" for data. The information derived from these systems is limited to the division that maintains control of the database. Little, if any, usable strategic knowledge can be derived from these systems without manpower intensive and time consuming analysis.

The databases considered for this thesis are TFMMS, RIMS (FM), WINPAT, Mandays, RTSS, and Global. Each of these databases has a specific purpose, which will be described in the following sections.

1. TFMMS

The Total Force Manpower Management System (TFMMS) is the single authoritative source for activity and manpower data for the US Navy. It consists of three major modules: activity, billet (requirement and authorization), and end-strength. TFMMS provides the ability to track manpower resources, requirements and authorizations for active military, officer and enlisted, reserves, civilians, contractors, and other categories of manpower. The information aids in defending Program Objective Memorandum and budget requests; and also supports recruiting, manpower and personnel management, personnel distribution, training, inventory management, and strength planning. This database is filtered for Naval Reserve information for use in this software development.

2. RIMS (FM)

The Reserve Integrated Management System (RIMS) Financial Management (FM) system is a Financial Management application created to manage those funds appropriated by Congress to the Naval Reserve. RIMS manages all detail documents in order to determine when to send accounting transactions to Standard Accounting and Reporting System, Field Level (STARS/FL). RIMS (FM) is a microcomputer-based application developed with the ORACLE programming language utilizing a Windows Graphical User Interface (GUI) environment and communicating with an Oracle database located on a central server. This database tracks documents that account for Naval Reserve Spending in the manpower/personnel arena.

3. WINPAT

The Windows Program Analyst's Toolkit (WINPAT) is the application software utilized to access the Department of the Navy (DON) programming Resource Allocation Database (RAD). WINPAT is used in the development of the Program Objectives Memorandum (POM)/Program Review (PR). WINPAT is currently used by N80, OPNAV staff, Marine Corps and Budget Submitting Offices (BSOs) to track decisions during the POM build. This database is filtered for Naval Reserve information and contains information on resource allocation.

4. Mandays

Mandays is a database that provides Reservist tracking data from a financial management perspective. It tracks funding for Naval Reserve orders for Annual Training (AT), Active Duty for Training (ADT), and Active Duty for Special Work (ADSW) as well as any other form of funded travel. The system provides feedback concerning support to the fleet through exercise and operations information tracking. It also tracks educational accomplishments through school attendance.

5. RTSS

Reserve Training Support System (RTSS) is a subset of the Reserve Headquarters System (RHS). RTSS provides billet information for the Naval Reserve. The database tracks information similar to that contained in TFMMS. However, the two databases do not have the same keys to identify the information. The RTSS database also tracks a much smaller number of data elements than does TFMMS. There is currently no automated update between TFMMS and RTSS.

6. Global

Global is a database extract from RTSS that provides specific manpower data related to each Naval Reserve member.

E. PROJECTS

Several projects have been initiated to assist with the accomplishment of one or more goals described in the "Leading Change" initiative. NaRSDAT is one such project. The goal is to coordinate the information from key Navy and Naval Reserve databases and to make this new information available to the decision makers. If cutting edge data warehouse and data mining techniques can be applied to the coordinated databases, a new, highly effective tool will be available for the strategic decision making process.

The knowledge developed through NaRSDAT should enable the strategic decision maker to make more timely decisions that are strategically focused. Reducing manual labor while providing a better product to the customer are basic requirements in establishing a 21st century force. When decision makers are allowed to spend more time evaluating and assessing, and less time gathering and organizing, an improved strategic decision making process is possible.

Through an object-oriented design approach, NaRSDAT will provide a modular, scalable, adaptable solution to the Naval Reserve Force for the assessment process. NaRSDAT will be the foundation on which an entire Reserve wide decision support system can be built. The use of object-oriented design methodology will ensure that NaRSDAT can be expanded to perform all the required functions in the foreseeable future. The next chapter describes an actor diagram and use cases as the first step in the object-oriented requirements analysis process.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ACTOR DIAGRAM AND DESCRIPTIONS

A. THE SYSTEM

NaRSDAT is the central system that will result from this development effort. The system is composed of a data warehouse and an on line analytical processing (OLAP) software application integrated by a Visual Basic application. These three elements work in concert to provide the functionality described in the background chapter. NaRSDAT receives input from the actors and carries out the requested operations or self generated operations. It will also produce the information and reports requested by the actors.

B. ACTOR DIAGRAM

An actor diagram depicts the actors and their interaction with the system. NaRSDAT has three actors: User, Administrator, and Legacy Database. These three actors are the only external sources of interaction with NaRSDAT. The actor diagram for this application is shown in Figure 3.1. As depicted, the Legacy Database actor has a one way interaction with NaRSDAT. All other actors have a bi-directional interaction with the system.

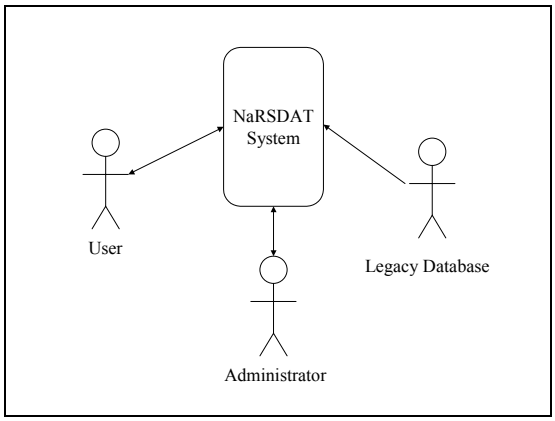


Figure 3.1 Actor Diagram

C. ACTORS

Actors are not part of the system. They represent persons or things that interact with the system. Actors are identified by asking questions about the system. In this case,

NaRSDAT has interaction with several actors. The following questions were posed to determine the actors for NaRSDAT:

- What will be the source of data for NaRSDAT?
- Who will utilize the system?
- Who will manage the system?
- What output will the system produce?

From these questions, three actors were identified. They supply the actions to answer the questions listed above. These actors are described in the following sections.

1. User Actor

The User actor is the common interaction with NaRSDAT. This actor will use the system on a daily basis. The User will request information and reports and request printouts of the supplied information. The User will input information about the general area of search, and refine the search for specific information. This input will be in the form of keyboard or mouse inputs. This interaction will only involve the OLAP portion of NaRSDAT.

A User will require different information from NaRSDAT based on the Department for which the User is working. A User from Manpower/Personnel (N1) will require access to a different subset of data than a User from Training (N7). This allows the general actor User to be broken into subcomponents: Training User, Manpower User, Assessment User, and Operations User. The relationship between the User actor and the four subcomponents is shown in Figure 3.2.

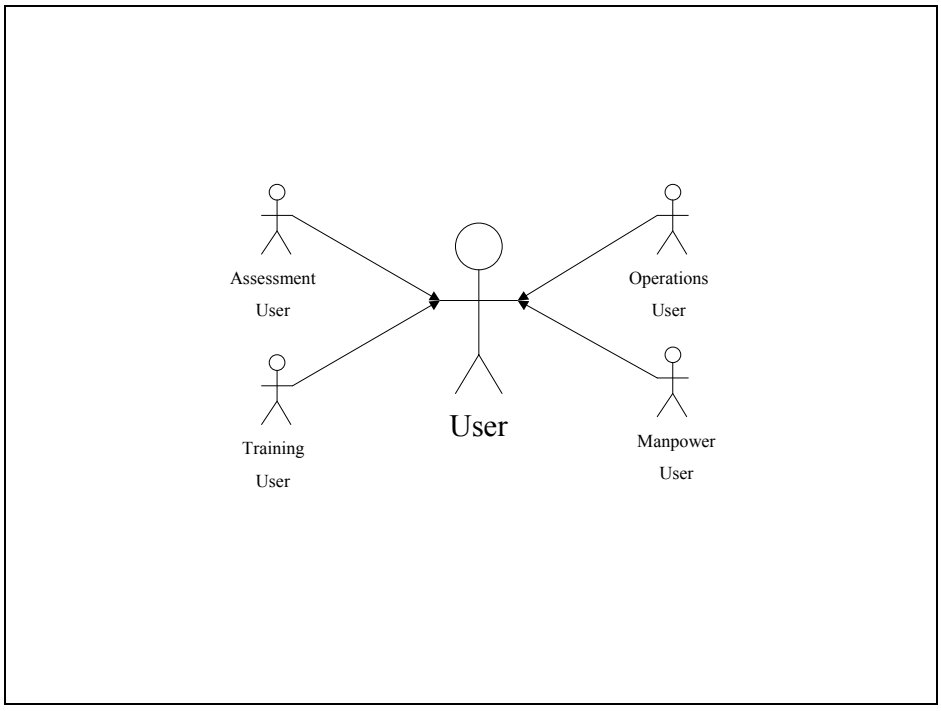


Figure 3.2 User Actor Components

This prototype will allow only local access to the system. The User will be at the local system and input directly to NaRSDAT. The User can be any individual with permission to access the system. The User will require previous knowledge of the general area of the search it wishes to conduct. The search can be refined through interaction with NaRSDAT, and will require a basic understanding of the NaRSDAT system.

2. Administrator Actor

The Administrator actor will have an oversight and maintenance interaction with NaRSDAT. The Administrator will place the legacy database files in the appropriate folder for use by NaRSDAT. This actor will also interact with the security section of NaRSDAT to allow designated Users to access the system. This will be done through the addition of User information to the security section of NaRSDAT.

This prototype will allow only local access to the system. The Administrator will be at the local system and input directly to NaRSDAT. The Administrator will be a designated individual with permission to access the system and provide maintenance and

security functions. The Administrator will require in depth knowledge of the NaRSDAT system to properly perform its roles.

3. Legacy Database Actor

The Legacy Database actor will interact with NaRSDAT in only one direction. It will supply the data required for NaRSDAT to properly fulfill its functions. NaRSDAT will request Legacy Database information periodically from designated locations. Once this task is accomplished, the Legacy Database actor will remain idle until the information is requested again.

There are six subcomponents that make up the Legacy Database actor. The TFMMS subcomponent supplies billet, activity, and end strength information to NaRSDAT. The RIMS(FM) subcomponent supplies financial management information. The WINPAT subcomponent supplies resource allocation information to the system. The Mandays subcomponent provides Naval Reserve financial management information to NaRSDAT. The RTSS subcomponent provides Naval Reserve billet, activity, and end strength information. Lastly, the Global subcomponent provides a subset of the RTSS subcomponent that is geared specifically for the Naval Reserve member. The relationship between the Legacy Database actor and the 6 subcomponents is depicted in Figure 3.3.

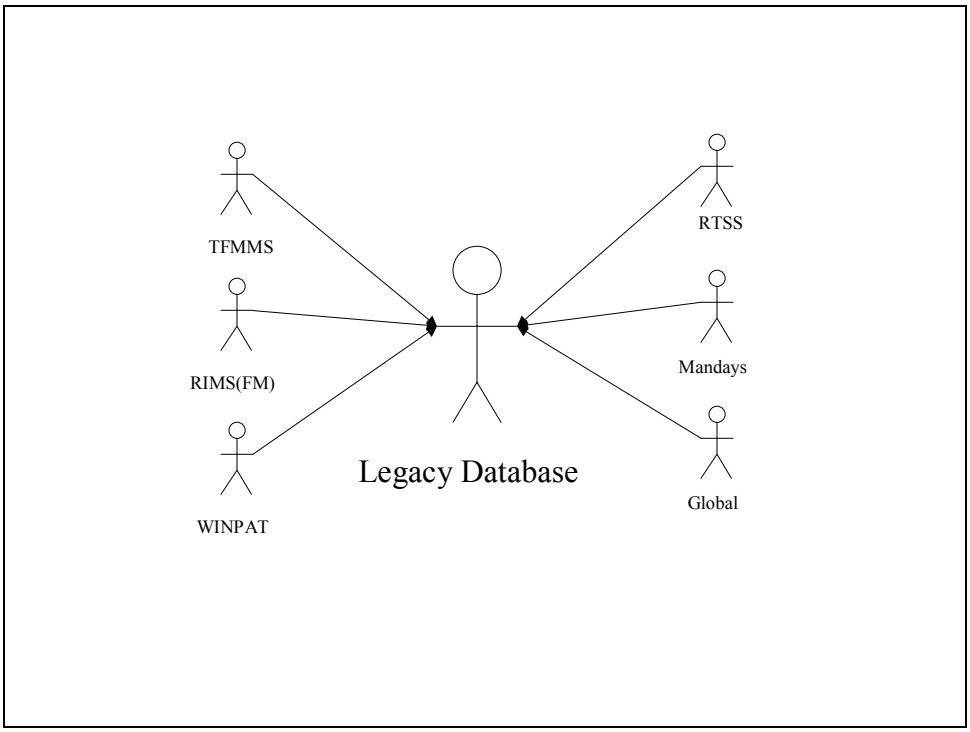


Figure 3.3 Legacy Database Actor Subcomponents

Since this prototype will only allow local access to the system, the Legacy Database actor must be stored on the local system. No system knowledge is required by this actor to carry out its function.

The interactions that actors have with the system are called use cases. Use cases help define system operation and functional requirements. Use case development will be explored in the next chapter, and the interactions between each pair of actors will be determined and defined.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. USE CASES

A. USE CASES IN UNIFIED MODELING LANGUAGE (UML)

UML employs use cases to assist in the development of functional requirements in system design. Use cases define the interaction between the actors and the system. They define the common tasks that an actor may perform during system operation. From the cases, a developer can determine the functional requirements for the system as well as any output required from the system.

Use cases consist of the execution elements that must be performed for an action to be carried out. A use case will describe the action to be performed and the actors involved in performing this action. It will also describe any conditions that must be met before starting the action and the state of the system when the action is complete. It describes, in detail, the steps through which the system must proceed to complete the action. In a properly defined use case, a developer can understand the condition of the system during all phases of the action.

1. Essential Use Case

UML incorporates two types of use cases. Essential use cases define an operation in generic terms. No technology, hardware, or software specific terminology or requirements are defined. At the initial level of development, this allows for a full exploration of the system while limiting vendor or personal biases. A developer's personal experience software, hardware, and technology can significantly influence a project. Although an essential use case is generic with regard to technology, it is still specific as to the actions to be completed. It must fully describe the beginning and end state of the system as well as the required steps to complete the action that is being performed.

2. Real Use Case

Real use cases are very well defined with regard to technology, hardware and software. Real use cases will provide platform and software specific steps to complete the requested task. In most developments, an essential use case will break into several real use cases. This is due to the complexity and length of the use case when technology

is applied to the situation. The real use case is described in such detail that the developer could sit at the system and actually carry out the operation.

B. ACTORS

Use cases use the actors developed in the actor diagram (figure 3.1) to initiate operations in the system. This section will develop the essential use cases for NaRSDAT. At the level of analysis that is currently being pursued, all six of the Legacy Database actors are carrying out the same actions through the system. Only the generic Legacy Database actor will be defined for the use cases in this chapter. The use cases for the individual Legacy Database actors would be identical to the generic use case that will be defined here.

The same logic applies to the User actor. There is no difference at this level whether the actor is the Manpower User or the Training User. All use cases defined here will be the same for each departmental user and the generic User actor.

The development of the real use cases, however, would involve applying the technology to the system and differentiating between the different sub actors for each generic actor shown in the actor diagram (figures 3.1, 3.2, and 3.3).

C. USE CASE DIAGRAM

The use case diagram is a visualization tool that aids in the development of the use cases. It uses standard UML icons and connections that visually indicate the actors and actions of the system. Figure 4.1 shows the use case diagram for NaRSDAT.

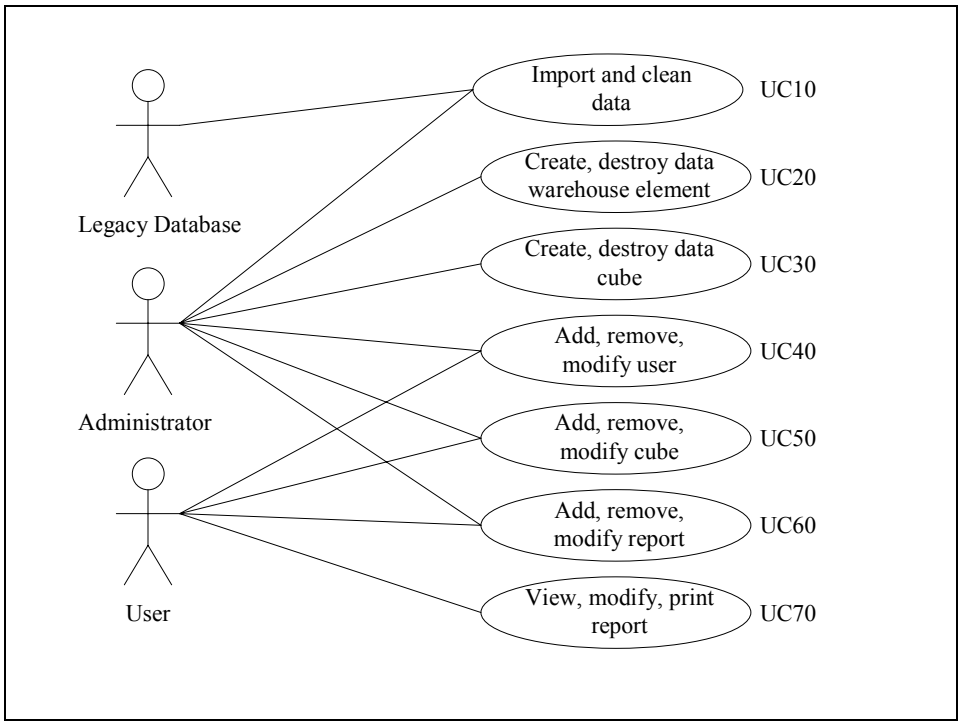


Figure 4.1 Use Case Diagram

D. USE CASES

A single actor can perform a task or several actors can be required to act in concert to complete the requested task. All elements of the required task are described in the use case. The specific use cases for each element shown in Figure 4.1 are developed in the following sections.

1. UC10: Import and Clean Data

The purpose of this use case is to import, clean, and format the data extracts from the legacy database systems. The Legacy Database actor and the Administrator actor are both involved in this use case. The only prerequisite for this use case is to have the data extracts pre-positioned in the designated directory on the local computer. The post condition of NaRSDAT will be a table or tables that are ready for the Create Or Destroy Data Warehouse Element use case. This use case will detect the presence of the data extracts and import the data. It will also clean the data and place the cleaned data in the appropriate table or tables for future use. It will then move the original data sources from the designated directory and store them in a backup directory. Exceptions will be noted if one or more data sources are not present in the designated directory. Exceptions will also

be noted if the use case could not clean the data or create the required table or tables in the designated database.

2. UC20: Create or Destroy Data Warehouse Element

The purpose of this use case is to give NaRSDAT the ability to create the required database schema to support the OLAP process. The Administrator actor is the only actor involved in this use case. There are two prerequisites for this use case. There must either be an imported and cleaned data source in the designated database to create a data warehouse element or there must be an existing data warehouse element in the database that is no longer desired or required. There will also be two possible post conditions from this use case. In the case of creating a data warehouse element, there will be a data warehouse element that is ready for use in the OLAP process. In the case of destroying a data warehouse element, there will be a compacted data warehouse available for use or backup. This use case will take the output from the Import And Clean use case. It will perform table manipulation and creation or destruction to provide a complete and compacted data warehouse for use in the OLAP process. The original database table or tables will be left intact for possible standard database query if the requirement arises. Exceptions will be noted if NaRSDAT could not create or destroy the desired data warehouse element.

3. UC30: Create or Destroy Data Cube

The purpose of this use case is to transform the data in the data warehouse element into a form that is usable by the OLAP interface of NaRSDAT. The Administrator actor is the only actor involved in this use case. There are two prerequisites for this use case. There must either be an available data warehouse element in the designated database to create a data cube or there must be an existing data cube in the data cube storage area that is no longer required. There will also be two possible post conditions from this use case. In the case of creating a data cube, there will be a data cube that is ready for use by NaRSDAT in the analysis and report process. In the case of destroying a data cube, there will be less but more relevant information available for analysis and reporting. This use case will take the output from the create data warehouse element use case or use an existing data cube. It will perform data manipulation and cube

creation or destruction to provide a complete and compacted data cube for use by NaRSDAT in the analysis and report process. Exceptions will be noted if NaRSDAT could not create or destroy the desired data cube.

4. UC40: Add, Remove, or Modify User

The purpose of this use case is to enable new users to access NaRSDAT, to change the ability of a current user to access data in NaRSDAT, or to remove a user from NaRSDAT access. The Administrator actor and the User actor are involved in this use case. There are three possible prerequisites for this use case:

- An existing user may require a different level of access to NaRSDAT;
- An existing user may no longer require access to NaRSDAT;
- A new user may require access to NaRSDAT.

There will also be three possible post conditions:

- A new user will have been added,
- An existing user will have been deleted,
- An existing user's permission will have been changed.

An existing user, a new user or the administrator will initiate this use case. A new user may request access to NaRSDAT through the administrator. An existing actor may request a change in their level of access to NaRSDAT. An existing user or the administrator may determine that a user no longer needs access to NaRSDAT. The administrator will then create, modify, or delete the user information in NaRSDAT. There are no exceptions for this use case.

5. UC50: Modify Data Cube

The purpose of this use case is to update the analysis and report section of NaRSDAT. The Administrator actor and the User actor are involved in this use case. There are three possible prerequisites for this use case:

- A new data warehouse element may be ready for use in the OLAP process;
- An existing data cube may contain information no longer be relevant to the analysis and report process,.

- An existing data cube may need to be modified to provide a different analysis and report capability.

The post condition for this use case is that a modified data cube will be available for analysis and reporting. In this use case, a user may need new information, modified information, or may no longer need information available in the analysis and reporting process. The administrator may also need to provide a cube to meet the requirements of another use case. The user or the administrator will initiate the use case by requesting the desired change. The administrator may either modify the existing data cube to meet the requirements, if the data is available, or may create a new cube by initiating use cases UC10 through UC30. The administrator will then incorporate the modified or new cube into the OLAP process. If the user requests a cube removal, the administrator will remove the data cube and confirm proper operation of the remaining data cubes. Exceptions will be noted if NaRSDAT could not modify the data cube because of insufficient data available. An exception will also be noted if the removal of information caused improper operation of NaRSDAT.

6. UC60: Add, Remove, or Modify Report

The purpose of this use case is to provide the user with a customized report capability. The Administrator actor and the User actor are involved in this use case. The prerequisite for this use case is that a user requires a change to the reporting capability of NaRSDAT. The post condition for this use case is that a new reporting capability will be available to the user. The user will initiate the use case by requesting a desired change to the reporting capability of NaRSDAT by the administrator. The administrator may modify an existing report, delete an existing report or create a new report. If the modification or creation requires data that is not currently available in the data cube, the administrator will initiate UC30 to obtain the required data in the format that is needed to provide the report. The administrator will then incorporate the modified or new report into the OLAP process. Exceptions will be noted if NaRSDAT could not add or modify the report because of insufficient data available.

7. UC70: View, Modify, or Print Report

The purpose of this use case is to provide end user functionality to the User actor. The User actor is the only actor involved in this use case. The prerequisite for this use case is that a user desires information from NaRSDAT. There are no post conditions for this use case. The user will initiate the use case by accessing NaRSDAT. The user will then request an existing report from NaRSDAT. NaRSDAT will present the requested report to the user. The user will have the opportunity to modify the report if required. The modifications will be limited to the user's permissions. When the user obtains the required presentation, he may print the report. At any time, the user may end this use case by terminating his access to NaRSDAT. Exceptions will be noted if NaRSDAT could not perform the requested modification to the report or could not print the report.

E. USE CASE SUMMARY

The use cases described in the previous sections are summarized in Table 4.1. The use cases from this chapter, in conjunction with user and administrator interviews conducted at CNRF, will be used in the next chapter to develop the functional requirements for NaRSDAT.

Use Case	Prerequisites	Post Conditions	Description	Exceptions
UC10: Import And Clean Data	Data extracts in proper location	Table(s) created	Detect data extracts, import and clean data.	Data source not present. Could not clean data. Could not create table(s)
UC20: Create Or Destroy Data Warehouse Element	Cleaned table available Existing element available	Revised data warehouse available	Detect table, create element Delete element	Could not create element Could not delete element
UC30: Create Or Destroy Data Cube	Available element Available data cube	Revised data cube	Manipulate element into data cube Delete existing data cube	Could not create or destroy cube
UC40: Add, Remove, Or Modify User	User needs change to current NaRSDAT access	Revised user access	Identify new requirement Change access privilege	None
UC50: Modify Cube	Existing cube needing modification	Modified data cube	Need for change identified Change executed	Could not modify cube Modification caused improper operation of NaRSDAT
UC60: Add, Remove, Or Modify Report	User requires change to report	New report capability present	Need for change identified Change executed	Could not add, modify, or delete report
UC70: View, Modify, Or Print Report	User requires information from NaRSDAT	None	User identifies required information User accesses information	Could not modify report Could not print report

Table 4.1 Use Case Summary

V. FUNCTIONAL REQUIREMENTS

A. FUNCTIONAL REQUIREMENTS IN UML

Functional requirements are used in UML to take an end user view of system development. Since the final product should meet all practical end user requirements, the system should be developed from the ground up to include these requirements. In the UML development approach, the end user's requirements play a key role in system definition. To this end, the developer must have a complete picture of what the user expects from the system.

The system developer should interview the users of the proposed system and explore any current systems for additional requirements that may not have been presented during the interviews. A list of products generated by the end user is also very helpful in determining a complete picture of system requirements. Many data elements that are not fully understood by the end user may become clearer when looking at the output products.

The end user may have locally or individually developed applications that are used to produce a site-specific product. These products may be more difficult to find, but are no less important than the globally available products. Many times, the locally developed applications may present a view of the available information that would benefit the entire organization. Due to corporate policy, personal goals, or geographic restrictions, this application, or reports generated by this application, may not be available to other corporate or departmental users. The inclusion of this information into the system development may increase productivity or assist in the strategic decision making process.

The functional requirements development process does not take into account system capability. Just because the system is capable of providing additional services or information does not mean that those functions or information are desirable. In a functional requirements development, this additional functionality can be applied to the existing problem and not wasted on peripheral activities. Whether the excess is CPU

cycles, system availability, or printer consumables, the surplus in system capacity can be directed to functions and processes that directly relate to the strategic goals of the company. This development strategy assures that the new system will do everything the user needs it to do and nothing else.

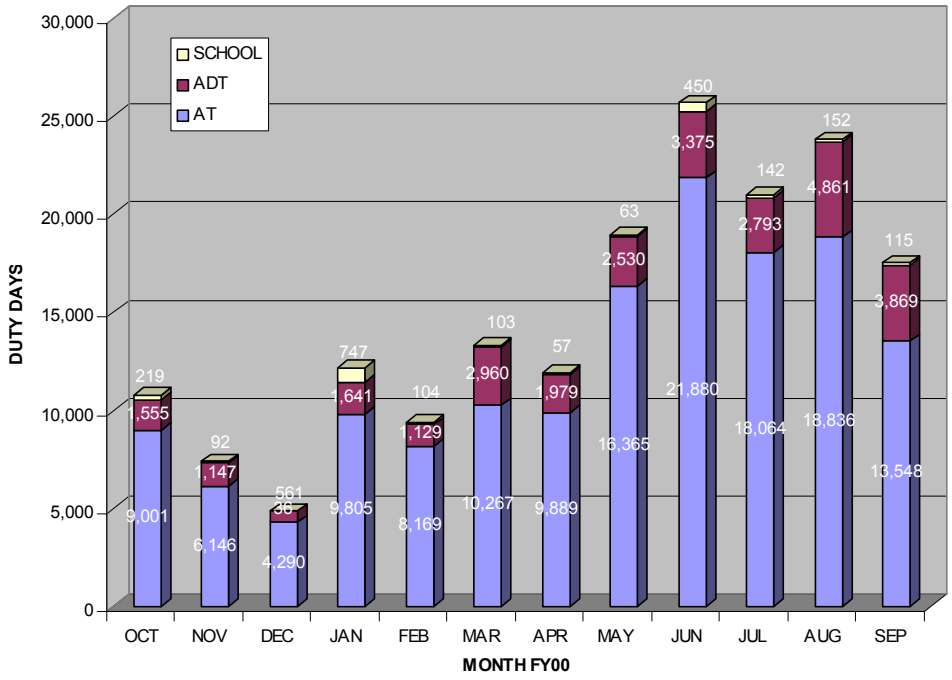
B. NARSDAT DEVELOPMENT

In developing the functional requirements for NaRSDAT, two primary resources were used: the CNRF Annual Assessment and the key performance metrics identified by N8. These two products are currently used to evaluate the Naval Reserve Force's performance and understand the impact of internal and external influences on this performance.

Due to data limitations and time constraints, the annual report deals primarily with financial indicators of performance. Other issues such as morale, training, and support to the fleet are more difficult to evaluate by hand. For this reason, personal interviews were conducted to determine the additional requirements for NaRSDAT.

Figure 5.1 shows an example of a specific report from the CNRF Annual Assessment that NaRSDAT must be able to provide. For additional reports, see Appendix B. NaRSDAT should be able to generate the typical graphic and textual information that is required by the end user.

Air Forces RPN by Month



FY-00 Review

-176,940 Duty Days were utilized in FY-00 by Air Forces

-146,260 days of AT; 27,875 of ADT; and 2,805 of ADT (Schools) were used.

-4th Quarter saw the highest usage for AT days (50,448, or 34%) and ADT (11,523, or 41%); and 2nd Qtr for ADT Schools (954, or 34%).

-The high AT month was June; ADT month, August; Schools, January.

-The monthly mean for AT was 12,188; for ADT 2,323; and Schools, 234.

FY-99/00 Comparison

-September had the largest gain in AT usage (5,777 days, or 77%). May and August showed small gains. The remaining months declined in AT usage. AT usage, overall, declined in FY-00 by 9%. 3rd and 4th Quarters deployed 67% of the AT days.

-ADT usage declined in FY-00 by 13%.

-ADT (Schools) usage declined by 3%.

Figure 5.1 Assessment Report Example

C. PERFORMANCE METRICS

From an analysis of the CNRF Annual Assessment and interviews with system end users, four categories of performance metrics emerged: RPN Execution, Manpower/Personnel, Mandays Provided, and Fleet Support. These categories cover internal and external customers, monetary expenditures, and military support. Each of these categories are broken into several specific performance metrics that will be supported by NaRSDAT.

1. RPN Execution Metrics

RPN execution metrics deal with how the Naval Reserve Force spent its money during the time period in question. The expenditures in question relate only to Reserve Program Navy (RPN) expenditures. For the Naval Reserve, this pertains to Selected Reserve orders for Annual Training (AT), Active Duty Training (ADT), Active Duty for Special Work (ADSW), and other similar expenditures.

RPN expenditures need to be evaluated by the type of orders (AT, ADT, ADSW, etc), by the Echelon III command (Air or Surface), by Program Code (Construction Forces, Air Forces, Medical Forces, etc), by rate or rank (YN2, LCDR, etc), and by Resource Sponsor (Atlantic Fleet, Chief of Naval Operations, Bureau of Medicine, etc)

Dividing each Echelon III command into its Echelon IV, V, and VI commands to evaluate expenditures would also be very valuable. This ability has not been demonstrated under the current system because of database conversion limitations.

2. Manpower/Personnel Metrics

Manpower/personnel metrics are concerned with overall manning for the Naval Reserve. End strength, a Congressionally mandated number of personnel permitted to be in service, significantly affects funding, Naval Reserve support, and recruiting. Manpower/personnel metrics are directly concerned with the number of personnel authorized and the number of personnel currently available.

The metrics evaluated for manpower personnel include end strength and billets. These metrics also include two different data sources, TFMMS and Global, and the comparison of the two sources. The metrics that NaRSDAT is required to support are end strength by fiscal year, end strength by paygrade (E4, O4, etc), billets by rate (YN3,

BM1, etc), billets by designator (1315, 1125, etc), billets by resource sponsor (Atlantic Fleet, Chief of Naval Operations, Bureau of Medicine, etc), and billet database comparison (Global vs. TFMMS).

3. Manday Metrics

Manday metrics are concerned with where each day of a Selected Reservist's time was spent. These metrics provide information on how many people and how many days of support were provided to each of our customers. This metric also provides information on which commands provided that support.

The metrics evaluated for mandays will cover statutory (by law) versus executed mandays for officer and enlisted, executed mandays by designator and by rate, percent performance by rate and rank, executed mandays by resource sponsor, executed mandays by type orders (AT, ADT, ADSW, etc), and executed mandays by month, quarter, and year.

This is one of the most important metrics for both internal and external customers. The metric provides information pertaining to legally binding minimums for a retirement year for a Naval Reservist. This information can be directly linked to morale and welfare issues within the force. It also provides an insight into what portion of the force is performing the majority of the work and what portion of the force is not performing. This can affect funding and end strength levels directed by Congress.

4. Support Metrics

The support metric deals with our external customers. This metric provides information about the number of Selected Reservists and mandays provided to support exercises, operations, and major claimants. This metric is divided into types of support such as AT, ADT, ADSW, etc.

Support metrics need to be evaluated by time period (month, quarter, year), by exercise, by program code, by rate, by designator, and by type support (AT, ADT, etc). The metric will be evaluated with both number of Selected Reservists and number of mandays.

This metric provides customer feedback as to the extent to which the Naval Reserve Force is meeting the needs of the fleet forces. It also provides justification for manning level increases or decreases in selected rates or designators.

D. ADDITIONAL METRICS

Other metrics are desirable for the complete evaluation of the Naval Reserve Force. These metrics derive their information from databases not currently in the NaRSDAT development. Information pertaining to retention, hardware, construction and any number of other topics relate directly to the overall performance of the NRF. These data sources and the accompanying metrics and requirements will be incorporated into future rollouts of NaRSDAT. One of the key benefits of a properly constructed data warehouse is that it is scalable. Future data sources can be rolled into the current warehouse with no impact on current operations and with an effort commensurate with the additional data.

The performance metrics of this chapter form the basis of the data model that will be developed in Chapter VI. The fact tables and dimension tables that make up the data model are built around the requirements of the performance metrics from this chapter.

VI. DATA MODEL

A. DATA MODELLING

The process of modeling data has been well studied for many years. There are several techniques available to model data. The two most relevant modeling approaches for NaRSDAT are relational modeling and dimensional modeling. The relational modeling approach will be used for normal database queries and the dimensional modeling approach will be used for the data warehouse portions of the system.

1. Relational Modeling

The two primary types of relational database modeling are Entity-Relationship (E-R) modeling and Semantic Object Modeling (SOM). E-R modeling defines meaningful data objects as entities that must be related to other entities for the model to be understandable. SOM defines meaningful data objects as semantic objects which are complete descriptions of objects that can be understood independently of other semantic objects in the model.

The decision as to which modeling process to select is influenced more by the modeler's training and experience than by technical requirements for the model. The NaRSDAT system will be developed using SOM since that is the modeling technique the author has the most experience with.

2. Dimensional Modeling

There is only one type of dimensional modeling. A dimensional model consists of dimensions and facts. A fact is directly related to a metric or performance parameter that is required to define or describe an operation of significance to the organization. A dimension more completely describes one of the attributes of a fact.

A dimensional model is not normalized. A normalized model, like SOM, minimizes the repetition of data to conserve space and make general queries more meaningful and swift. Dimensional models intentionally duplicate data in the fact table to speed the data mining process.

B. SEMANTIC OBJECT MODEL

The SOM attempts to model the end user's view of the data. This makes development and user integration more successful. An object in SOM must be described completely. This is done by describing the object, its characteristics, and any relationships to their objects within every object. Unlike the E-R model, which shows relationships through line diagram connections, the object in SOM includes characteristics for the relationships with other objects in the model.

The defining characteristics of an object are called attributes. Attributes can be simple, group, or semantic object attributes. A simple attribute is a single, independent characteristic. A group attribute is a single concept defined by several attributes that are related. A semantic object attribute is an attribute that establishes a relationship with other objects in the model.

Objects also have identifiers. Identifiers are similar to primary keys in a database. They uniquely identify an instance of an object. There are two types of identifiers. An object identifier is a simple attribute that uniquely identifies an instance of an object. A group identifier is a group attribute that uniquely identifies an instance of an object.

Each attribute of an object can either be required or not required in a specific instance of the object. Each attribute can also be repeated once or many times. The minimum and maximum number of times an attribute can be associated with an object is known as cardinality. A model will always define the minimum and maximum cardinality of each attribute listed in the object.

Using a graphical interface of boxes, which represent objects, and attributes, identifiers, and cardinality, a complete SOM can be described. Figure 6.1 shows a generic SOM. This model shows two objects related with a semantic object attribute. It also shows all other elements of the SOM described in the previous sections.

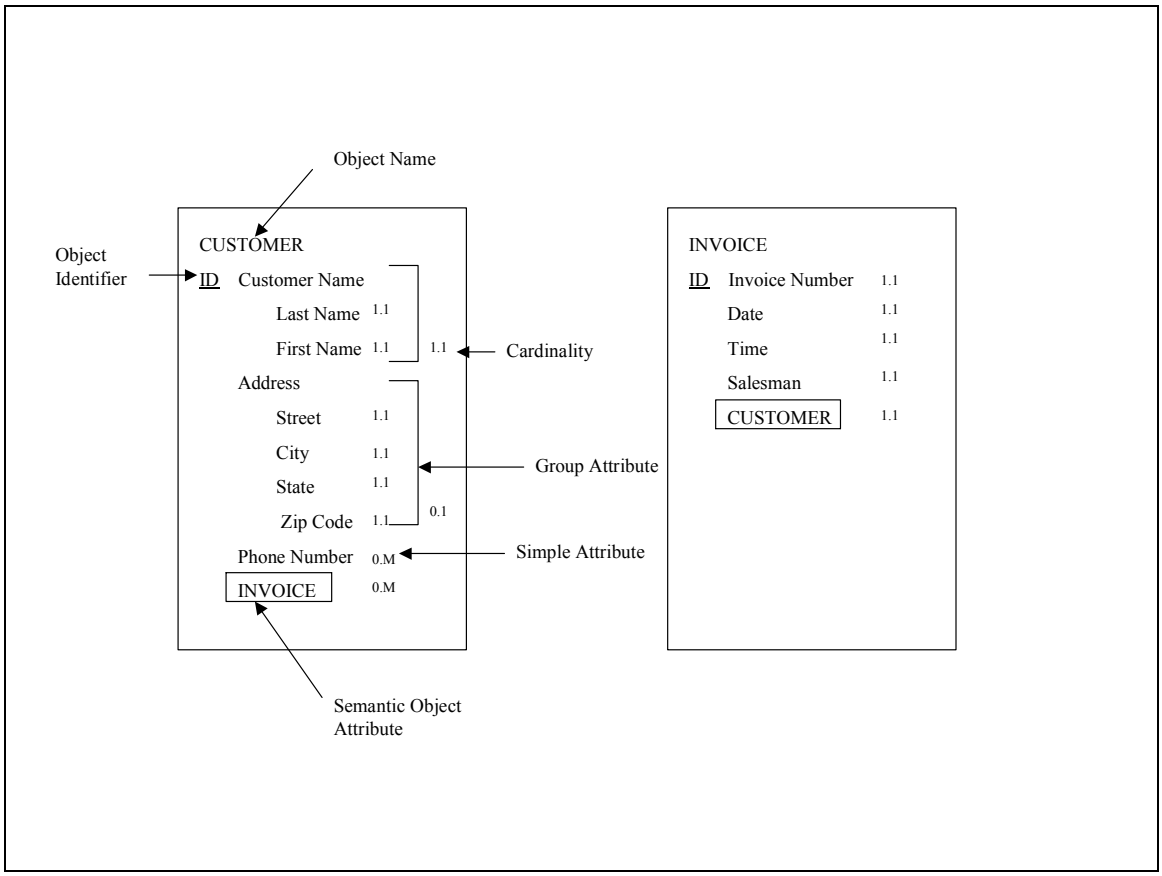


Figure 6.1 Semantic Object Model Example

C. NARSDAT SEMANTIC OBJECTS

The NaRSDAT data model consists of six semantic objects: Member, Orders, Billet, Active Unit, Reserve Unit, and TFFMS Billet. These objects provide complete functionality for NaRSDAT and provide all data fields needed for the dimensional model and for the data mining process. The next sections will describe each object and then provide a graphical SOM of the system

1. Member Object

The Member object describes a Naval Reservist. The object identifier will be the Social Security Number (SSN). This object contains such information as name, address, rate, grade, and several other attributes. There are three group attributes. They are address, consisting of street, city, state, and zip code, NEC, consisting of PNEC and

SNEC, and IAP, consisting of IAP status and IAP date. The Member object contains two semantic object attributes; Orders and Billet. Figure 6.2 shows the complete object.

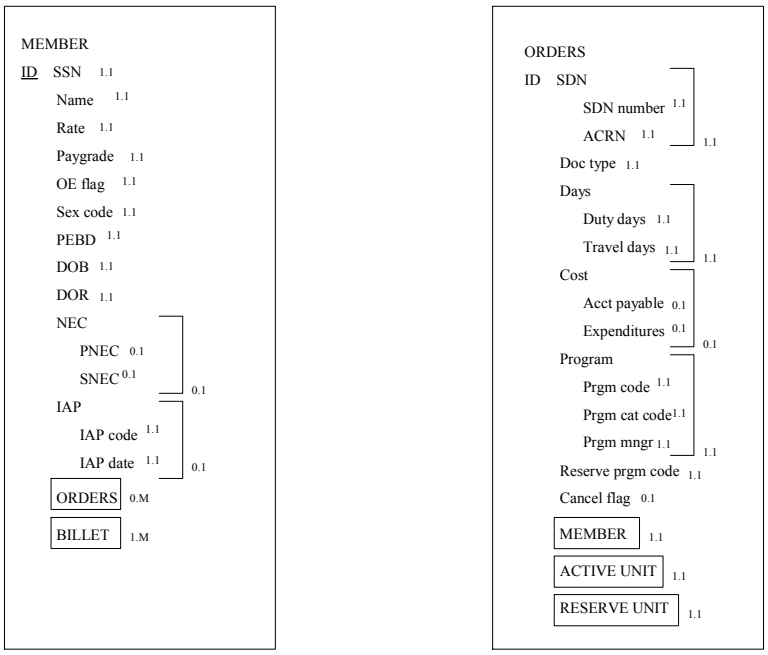


Figure 6.2 Member and Orders Objects

2. Orders Object

The Orders object describes a set of orders that obligates Naval Reserve funds. The object identifier is a group attribute. The group is SDN, which is made up of SDN number and ACRN. Some simple attributes are fiscal year, document type, PMC, and RPC. Other group attributes are days, consisting of travel days and duty days, expense, consisting of accounts payable and expenditures, and program code, consisting of program code and program category code. This object also has Reserve Unit, Active Unit, and Member as semantic object attributes. Figure 6.2 details the complete object.

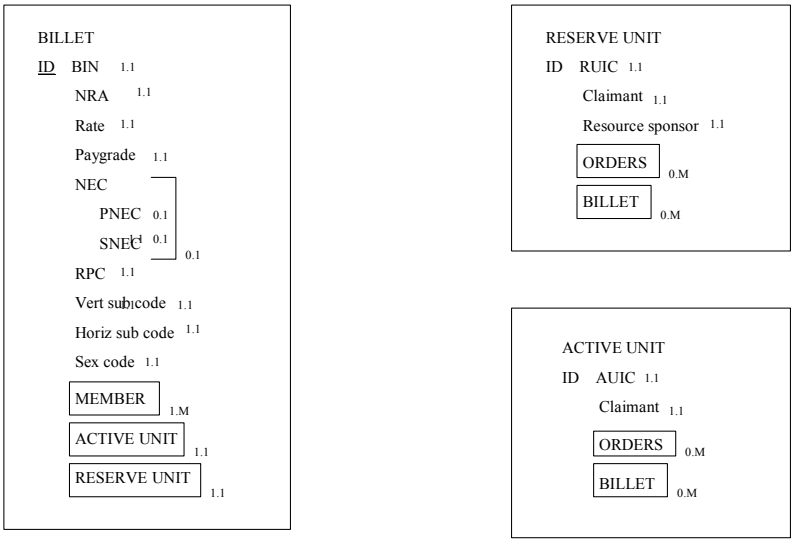


Figure 6.3 Billet, Reserve Unit, and Active Unit Objects

3. Billet Object

The Billet object describes a job that a Naval Reservist fills. The object identifier is the Billet Identification Number (BIN) attribute. Some simple attributes are Naval Reserve Activity (NRA), Rate, Paygrade, and Reserve Program Code (RPC). The billet object has one group object, NEC, which consists of PNEC and SNEC. This object also contains three semantic object attributes. They are Member, Active Unit, and Reserve Unit.

4. Reserve Unit and Active Unit Objects

The Reserve Unit and Active Unit objects are simple objects that have the Unit Identification Code (UIC) as the object identifier. Each object has the Major Claimant as a simple attribute. The Reserve Unit object also has the Resource Sponsor as a simple attribute. Each of these objects contains semantic object attributes of Billet and Orders.

5. TFFMS Billet Object

The TFFMS Billet object is a standalone object. There is no direct link to the other objects in NaRSDAT. This object is used solely for a rough comparison of billet

information between the Naval Reserve and the Fleet Navy databases. This object has simple attributes of Major Claimant, Resource Sponsor, Designator, Grade, Rate, and AUIC. Since several attributes have different meanings in the two databases and no one-to-one mapping is possible, only general comparisons will be accomplished with this object.

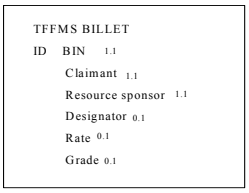


Figure 6.4 TFFMS Billet Object

D. NARSDAT DIMENSIONAL MODEL

The NaRSDAT dimensional model is developed through the use of a star schema. The fact table is the center of the star schema. The dimension tables form the points on the star. Unlike the relational model, which directly links to tables in the relational database, the dimensional model does not link directly to any tables in the database. The model is a logical construct that assists in properly configuring the data warehouse section of NaRSDAT.

Each fact table is based on one of the four metrics from chapter five. The fact table provides the basis for the data mining operation. The dimension tables associated with that fact table provide the specifics for the reports and graphical output. The fact table contains an identifier from each of the dimension tables as well as other attributes specific to the fact table. Each dimension table can be linked logically to more than one

fact table. A dimension table of time is linked to every fact table. This table is what enables the unique data mining capability of the software.

1. RPN Execution Star

The RPN Execution star is built around the RPN Execution fact table. It contains identifiers from the following fact tables: Time, Order type, Claimant, Echelon 3, Program code, Rate/designator, Rank, and Resource sponsor. The graphical representation is shown in figure 6.5.

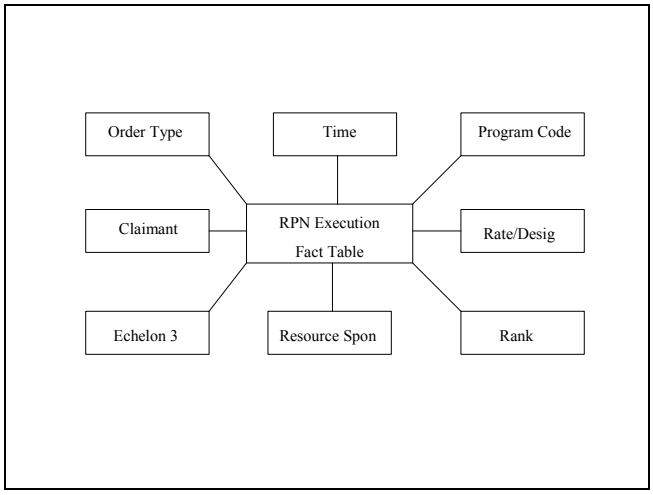


Figure 6.5 RPN Execution Star Schema

2. Manpower/Personnel Star

The Manpower/Personnel star is built around the Manpower fact table. It contains identifiers from the following fact tables: Time, Claimant, Rate/designator, Rank, and Resource sponsor. The graphical representation is shown in figure 6.6.

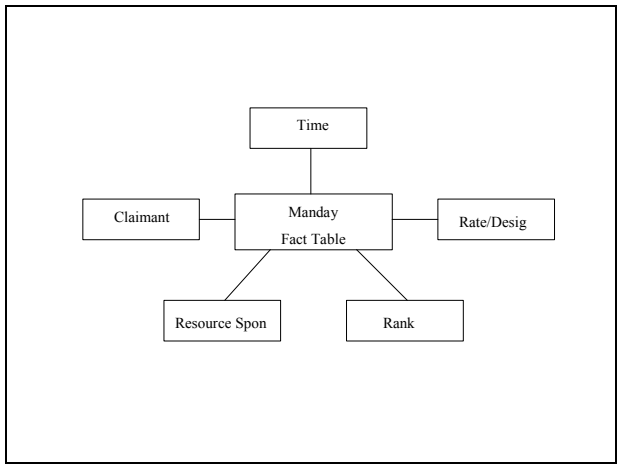


Figure 6.6 Manpower Star Schema

3. Manday Star

The Manday star is built around the Manday fact table. It contains identifiers from the following fact tables: Time, Order type, Claimant, Program code, Rate/designator, Rank, and Resource sponsor. The graphical representation is shown in figure 6.7.

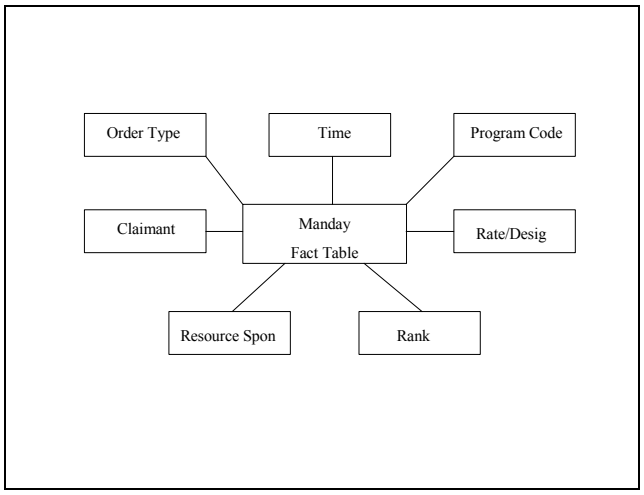


Figure 6.7 Manday Star Schema

4. Support Star

The Support star is built around the RPN Execution fact table. It contains identifiers from the following fact tables: Time, Exercise, Order type, Claimant, Program

code, Rate/designator, Rank, and Resource sponsor. The graphical representation is shown in figure 6.8.

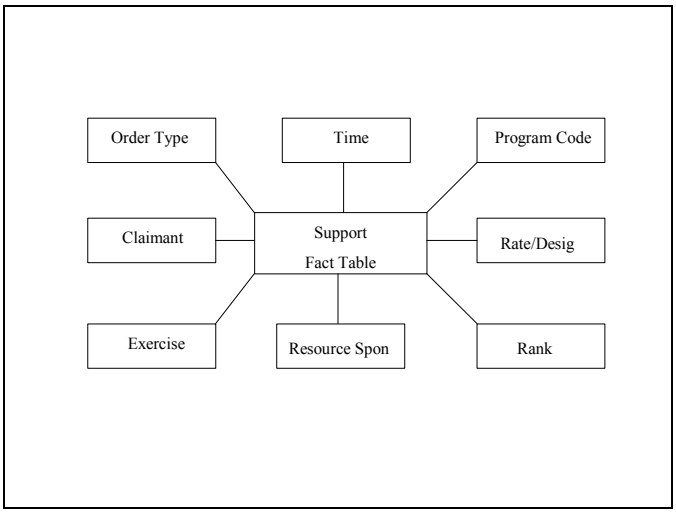


Figure 6.8 Support Star Schema

E. IMPLEMENTATION

With the database modeling complete, the implementation process can begin. The implementation is conducted in Microsoft Access™ for the data warehouse process and Cognos Powerplay™ for the data mining process. The last major concern for implementation is the data quality and import process from the legacy databases. This will be covered in the next chapter.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. DATA MIGRATION REQUIREMENTS

A. MIGRATION ELEMENTS

Data migration consists of three major processes in data warehouse development: extraction, cleaning, and importing. Each process has unique problems and solutions depending upon the data warehouse being constructed. The data migration process is perhaps the most critical success factor in the entire data warehouse project. The timeliness, quality, and structure of the data being imported will establish the quality, speed, and functionality of the entire data warehouse.

B. EXTRACTION PROCESS

The extraction process can be accomplished in two ways. Data can be manually or automatically downloaded to a file or disk, which is then transmitted to the data warehouse via mail, e-mail, or network links. Data can also be accessed directly by some form of connection between the data warehouse and the legacy database. This type of data extraction can be automated and provides a much more flexible means of data acquisition. However, this type of data extraction also tends to cause more problems since, in most organizational structures, data is directly related to power and influence. Those who control the data may not want others to have unfettered access. The security requirements for direct network access to a legacy database can also pose problems with implementation of an automated extraction system.

C. CLEANING PROCESS

Of paramount importance during the data migration process is data quality. If the quality of the data being inserted into the data warehouse is not good, the resulting information will also be of questionable quality. There are several areas of interest in the

cleaning process: standardization, duplication, splitting, merging, incorrect data, and missing data.

1. Standardization

The data warehouse developer must first ensure that all data is entered in a standard format. The developer can easily define the metadata for the data warehouse. However, ensuring that the transformation from legacy data to warehouse data can be difficult. The administrator must rigorously test the legacy data for errors and continually watch the fields for changes of data type, changes in format or complete removal.

2. Duplication

The data may have duplication that must be removed. This can be an extremely difficult process. For example, in two legacy databases, a data element may contain an individual's name. In one of these databases, the name may be defined as first and last name, whereas in the other database, the name may be defined as first and last name and middle initial. This conflict must be resolved through a testing process that can identify duplicates.

3. Merging or Splitting Fields

The source data may need to be transformed and combined. For example, a database may contain address information. In the legacy database, the address may be divided into street number and street name, whereas, in the data warehouse, the requirement may be to have only a single text string for street address. Identifying this data and applying the correct transformation as it is entered into the data warehouse is essential.

The opposite situation may occur as well wherein data fields may need to be split. For example, an address or phone number entered as one piece of data in the legacy

database may need to be decomposed in the data warehouse so the user can group information by zip code or area code.

4. Incorrect Data

Data quality is also affected by corrupt data or data entered incorrectly. A good data extraction tool will provide testing and remediation routines for dealing with this kind of problem. Although some tools allow for automatic repair of this kind of problem, most data warehouse administrators will wish to know when these type problems occur. Most tools will either print a report of data quality errors or will display an interactive user interface to assist the administrator in repairing these errors.

5. Missing Data

Missing data is a particularly interesting and vexing problem. Some missing data can be entered as a standard flag with fields generated to annotate the error involved. Some missing data can be tested against generic data and an appropriate entry (e.g., the mean) made in the data warehouse. Other missing data must be identified to the data warehouse administrator for corrective action. Particular attention must be paid to data elements that become identifying, or key, attributes for the data warehouse. If these elements are missing, the importing process would generate data warehouse errors and could cause interruption in the operation of the data warehouse.

D. IMPORTING PROCESS

Once the data has been extracted from the legacy database and cleaned, it is ready to be placed in the data warehouse. The import process involves testing the data warehouse elements for duplicate entries and inserting all data that is not a duplicate. It also involves placing the data in the appropriate table in the correct format. The testing process can be accomplished prior to extraction from the legacy database if a live

connection is used for the data extraction. If not, duplicate testing must be performed at the data warehouse by comparing each imported element against the entries in the existing database. The import process will usually be the most time-consuming part of the data migration process. The automation of the import process is of significant importance. When the process is automated, it can be done during predetermined slow times of operations so as to minimally impact system performance. NaRSDAT has elements of each of the three components of the data migration process.

E. NARSDAT EXTRACTION

The extraction process for NaRSDAT is currently a manual one. Each legacy database operator extracts a portion of the database to a file. These files are e-mailed to CNRF N8. The extracted files are formatted according to a standard Naval Reserve analysis requirement and have consistent fields upon arrival at N8. Although this is a relatively time consuming solution to the extraction process, obtaining an automated extract through a direct connection to the legacy databases is beyond the scope of this thesis. An automated extract involves placing a stored procedure on the system that contains the legacy database. This stored procedure then activates at designated times and transfers the data. Usually, the transfer is performed to a staging system that cleans the data in preparation for entry into the data warehouse. Effort must be exerted to ensure the consistency of the extracted product. NaRSDAT uses predefined specifications for each import. If the fields or field formats change, the specification will require manual correction to perform properly.

F. NARSDAT CLEANING

The NaRSDAT cleaning process will be accomplished through a Microsoft Visual Basic interface. This interface allows the data warehouse administrator the opportunity

to identify the files to be cleaned and imported. Once the files have been identified, the cleaning process begins.

1. Working With Legacy Files

The first requirement is to place the legacy database files in a format that will allow for proper manipulation prior to migration. With the NaRSDAT development, this is performed by importing the files into tables in a Microsoft Access database that have been formatted for the specific type of data. The tables are predefined and supplied as part of the NaRSDAT installation process.

To properly import Comma Separated Value (CSV) text files, a specification must be defined in Microsoft Access. This specification can be saved and referred to in the Visual Basic application. The specification allows the program to define the fields available, the format for each data field, and any fields to be skipped in the import process. An interesting by-product of the automated process in Access is the error table that is generated by the import process. This table shows the row number, the field label and the error type for all import errors generated by the import process. This table is available for reporting the import errors to the data warehouse administrator.

2. Formatting the Data

The next step in the cleaning process is to ensure that the data elements are formatted to the requirements of the data warehouse. The elements that make up the primary keys in the Access relational tables and fields that equate to performance measures in the PowerPlay cubes are the most significant. The cubes involved in the PowerPlay system are database tables arranged in a star schema as discussed in the

previous chapter, and based upon time and other relevant metrics. Once the associated dimensions are defined, the three dimensional data structure becomes a cube.

There are only two significant formatting issues with the legacy database fields that are handled in NaRSDAT. The first is the Reserve Unit Identification Code (RUIC). In Global and RTSS, a RUIC is five characters in an alphanumeric format. In Manday, the RUIC is defined as a 6 character alphanumeric entry. The Manday database defines a Naval Reserve unit with a preceding "R", and active duty units with a preceding "N".

To properly relate the information from the tables, a decision must be made regarding a standard format. We have elected to use the five character entry as the design standard. An Access query is implemented that removes the first character from the Manday RUIC entry and replaces the data in the table.

The other concern with the formatting process is the name of the member. Global uses a four character name for the member while Manday uses a five character identifier. Since the SSN is the primary identification for the member, we have elected to use the four letter name and disregard the name field in the Manday database.

3. Duplicate Data Removal

During the import process, NaRSDAT uses Access append queries to perform the data insertions. The primary keys, referential integrity and join types defined in the Access database prevent the addition of duplicate data during the import process. The removal of duplicate entries is automatic and no warnings or errors are received by the administrator. Since the legacy database extracts are received in the same format with all relevant data duplicated in each extract, there is no way to remove duplicates prior to the

import phase. The update query seems to be the most efficient method of removing duplicate entries.

4. Merging and Splitting Data

NaRSDAT required no merging or splitting data during the development.

5. Incorrect Data Remediation

Incorrect data in NaRSDAT is very difficult to handle. Since most fields are alpha or alphanumeric entries, there are no easy ways to check for correct data entry. The basic test of data length is performed automatically by the Access database. The referential integrity requirement also assists with ensuring correct data entry for the primary keys and join fields.

The primary keys for each of the relational tables are available from multiple data sources. For example, a RUIC is listed in Global as well as Manday and RTSS. Comparing the relational table entries against the original legacy databases will generate a report that can identify single use entries. Since an error in the key field would show up as a single entry, these reports assist in identifying incorrect data.

For attribute fields, no effort is made in this prototype to correct inaccurate data. The capability to perform this is available through lookup tables for several of the fields. Examples of this are the rate, rank, state, and claimancy. However, the performance of NaRSDAT is significantly affected on the prototype platform if these are implemented. The data errors in the legacy databases are not a significant problem. Most fields come from either standard entry forms or from drop down selection boxes. Errors in name, address or other personal fields are of no importance to NaRSDAT. Other fields such as document type, resource sponsor, or budget category are significant but are also easily

identified. The PowerPlay reports show blank or single entry fields for this data very well as a part of the normal report generation process. The incorrect data entry problem seems to be most significant in the primary key entries, since these fields are the ones that are manually entered routinely during data entry in the legacy database. The correction for any of these errors listed above would be a manual effort by the database administrator.

6. Missing Data in NaRSDAT

The only missing data field that is corrected in NaRSDAT is the Billet Identification Number (BIN). Approximately twenty five percent of the billets identified in the Global database do not have BINs associated with them. NaRSDAT contains a subroutine for identifying these billets and creating a unique BIN for that entry. An Access query is implemented that identifies the billets without a BIN, generates a unique number to identify that billet, and replaces that billet number in the legacy table.

Other missing data in NaRSDAT is actually used in the decision support process. Missing field information for such key areas as type of orders, exercise support, or active unit can assist N8 with evaluating the effectiveness of the field data entry procedures and policy changes.

G. NARSDAT MIGRATION

The NaRSDAT migration process involves moving properly formatted and cleaned data from the legacy database tables to the Access relational data warehouse tables. The first step in providing the migration path is to define the relational tables that will make up the data warehouse and the source tables for each of the fields. Then the migration process can be described.

1. Access Relational Tables

The Semantic Object Model is described in full in section 6.C. To implement this model, a relational table design must be developed. Using the Semantic Object Model and a standard relational table development scheme, the design of figure 7.1 was derived.

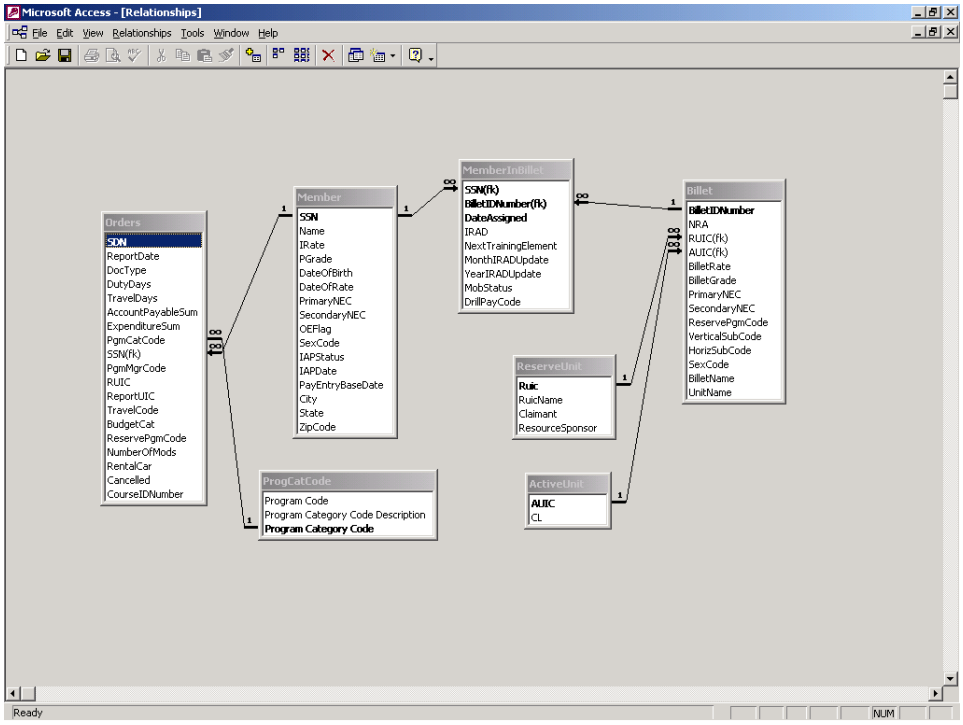


Figure 7.1 Relational Table Model

A complete data dictionary for this schema is provided in Appendix C.

The primary difference between this table design and the SOM is the use of the MemberInBillet table. A many-to-many relationship in a SOM must be implemented as a join table in a relational database. Also note that the one-to-many relationships are implemented as foreign keys in the table which contains the "many" portion of the relationship.

Of significant note in the design is the lack of a relationship between the ReserveUnit RUIC and ActiveUnit AUIC primary keys and the Orders table. Significant money is spent on units that are not recognized by Global or Manday as legitimate Naval Reserve units. The cause for this problem is not known but will be brought to the attention of the data warehouse administrator through report generation. Due to the amount of money and the repeat consistency of RUICs and AUICs involved, this data is still included in the data warehouse, however the relationships between the tables were severed. The same problem is encountered with the Member and Orders tables related on the SSN of the member. However, this problem can be attributed to the fact that many Naval Reserve members have left service prior to this download. Since there is no data memory in the extracts, the orders for these members will temporarily show up as money spent on members that are not in the Naval Reserve.

2. Migration process

The migration process is implemented through the use of Access queries in a pre-specified order. The queries were developed and tested on a sample database in Microsoft Access. The queries are then called and the associated tables filled in the following order in the Visual Basic application:

1. The peripheral tables for Reserve Unit and Active Unit;
2. Member and Billet tables;
3. MemberInBillet table;
4. Orders table.

This order is required to support the referential integrity designed into NaRSDAT.

There is one additional consideration with the migration process. The money for each SDN in the Orders table has to be calculated. The RIMSFM legacy database separates the SDN by ACRN. This is insignificant for data warehouse purposes, but requires a calculation to retrieve the proper information. This single import process requires three queries to migrate the data. First, a query must be performed to sum the money fields across an SDN. Next, a table must be made to hold this data. Last, the data must be updated to the orders table based on the SDN. The SQL code for these queries is listed in Appendix E.

H. IMPLEMENTATION

The next topic for discussion is how OLAP capabilities are implemented in NaRSDAT and how N8 will use the overall system. The next chapter will discuss the NaRSDAT user interfaces and how the PowerPlay software provides powerful OLAP interfaces and capabilities.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. PROTOTYPE IMPLEMENTATION

Once data migration has taken place from the operational data sources into the Access data warehouse as described in the previous chapter, the system is then ready for display and report generation. This aspect of NaRSDAT is facilitated by OLAP software provided in this case by Cognos™.

A. COGNOS POWERPLAY DEVELOPMENT

1. Data Cube Models

The Cognos system uses models to identify the structure of the data cube to be deployed. By defining the model, updated cubes can be developed without having to perform any further operations on the star schema data. The Cognos application for modeling and deploying data cubes is called Transformer. Transformer uses a standardized interface for all data cube modeling. An example of the user interface is provided in Figure 8.1.

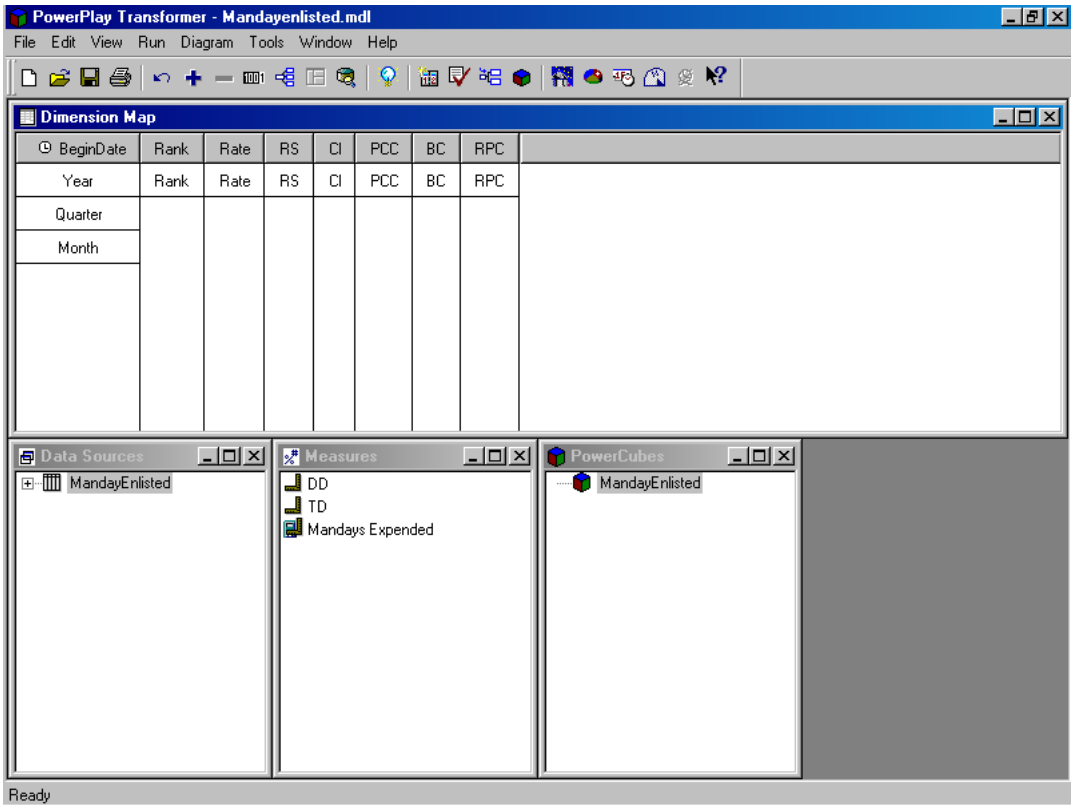


Figure 8.1 PowerPlay Transformer User Interface

The Transformer interface shows the key elements of the data cube design. The first entry required is the data source. The data source for NaRSDAT is the star schema tables from the data warehouse. The data source opens with an element for each field in the table. The next section of the model is the Dimension section. Dimensions are the areas in which the user may want to evaluate the data by rolling up or drilling down. Dragging the data source elements to the dimension header bar forms dimensions. Each of these dimensions will be available for user analysis in the PowerPlay cube and report section. The last area of the user interface is the Measure section. This portion of the interface is used to establish the metrics that will be used in the presentation of the data. Measures can be defined as a field of the source table, can be calculated, or can summarize or perform statistical operations on available fields. Once each of these interface sections is complete, Transformer generates categories based on the dimensions and measures provided and builds the data cube.

When the model is completed and saved, the model can be accessed from the Visual Basic code of NaRSDAT. Category generation and data cube creation are methods that can be performed by the Cognos object. By this method, the process of developing the data cubes can be automated from previously defined data cube models.

2. Exploring Data Cubes

Once the cubes are developed, the user is free to explore the data with PowerPlay. The explorer interface opens by displaying the most generic form of the data cube. Only tables are provided by default. The basic PowerPlay Explorer interface is shown in Figure 8.2.

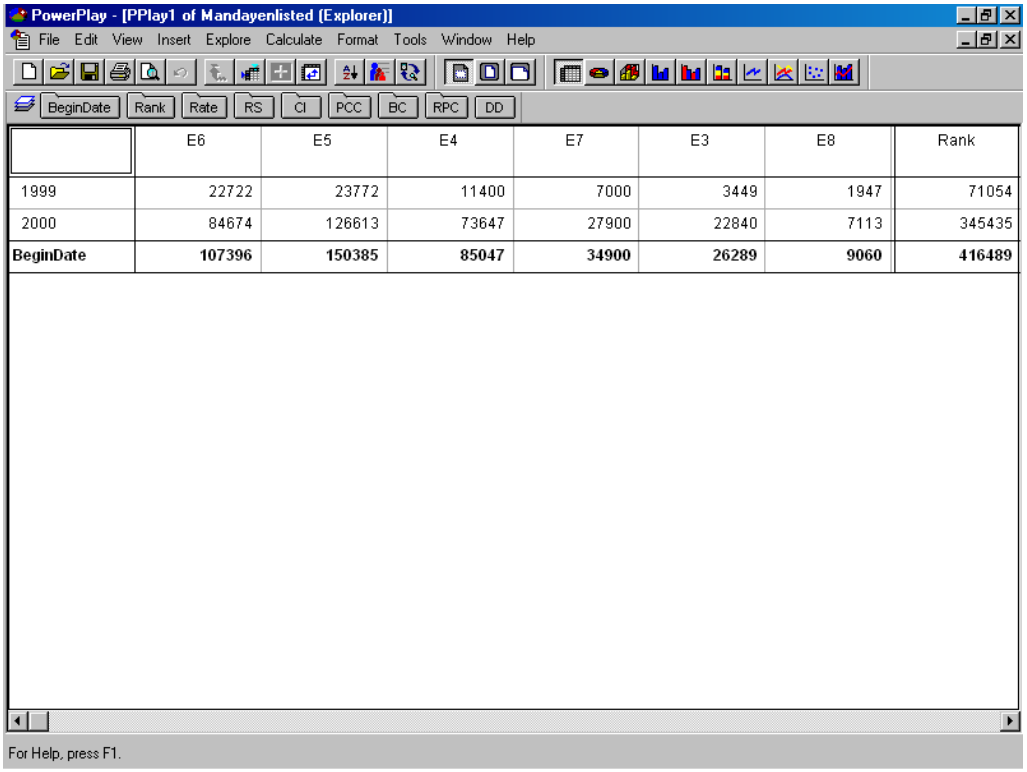


Figure 8.2 PowerPlay Explorer Interface

From this interface, the user can directly interact with the display of the data cube. All dimensions from the model are displayed at the top portion of the interface. Each of the dimensions can be dragged to either axis of the table and be displayed as the row or column respectively. Data are regenerated automatically to instantly fill in the table as dimensions are dragged and dropped to the appropriate axis. The user can also insert graphical displays of the data. The insert menu allows the user to place any of several types of graphs in the displayed area. Once the display is designed to the user's preference, he can save the display as a PowerPlay Report. Once saved in this manner, the report can be reopened and viewed at a later date with current data cube information already formatted to the user's preferences. The location for saving reports and the report format are predefined in the data model for that cube. Any new reports that are generated are saved, by default, in the specified location. This eases the developer's requirements to segregate the reports. An example of a formatted PowerPlay Report is shown in Figure 8.3.

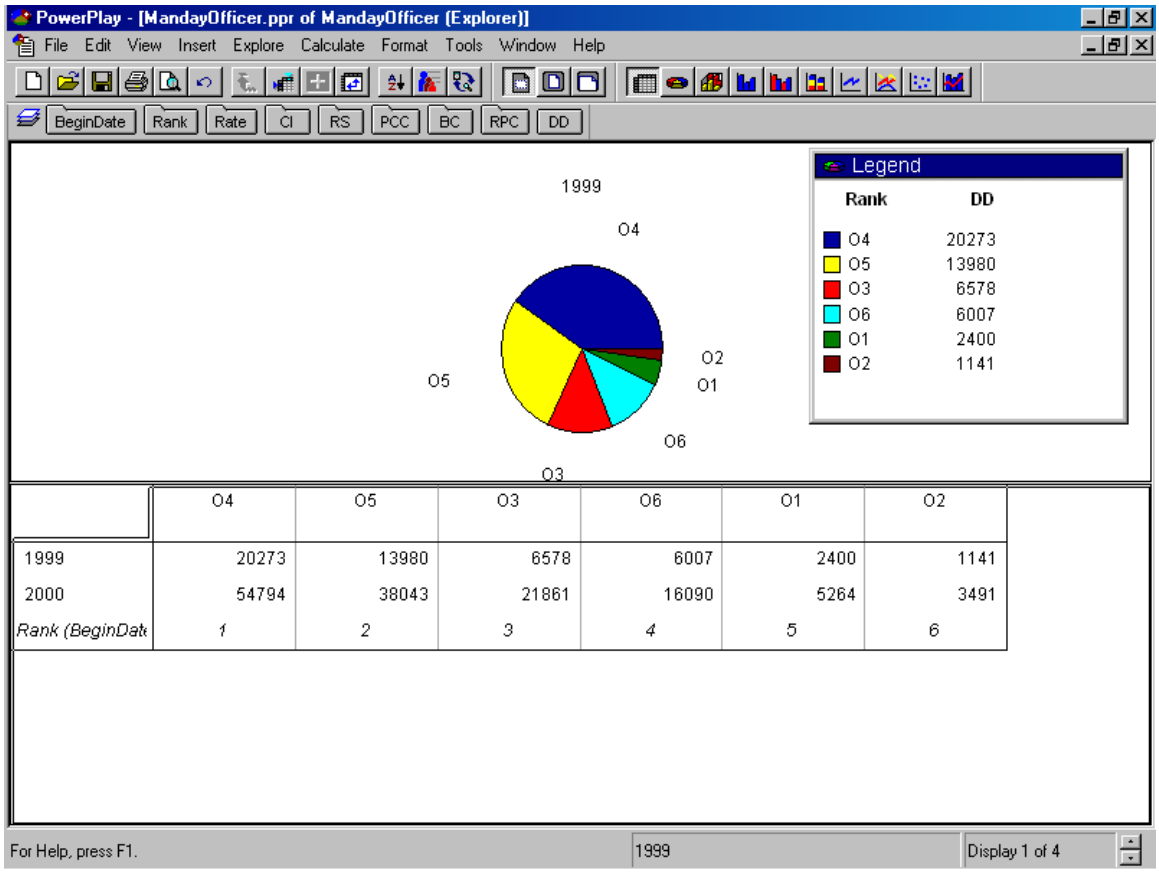


Figure 8.3 PowerPlay Report Interface

Figure 8.3 shows a typical report. The report shows mandays spent by rank for the officer force in fiscal year 2000. The top six ranks are shown and ordered by number of mandays spent. The ranking can be modified to include greater or fewer ranks and can also be ranked by the horizontal axis information. The inserted graph at the top displays the same information in the table display at the bottom of the report. As the table information is changed, the graph automatically changes to display the new information. A legend is supplied to the right of the table. The legend shows the categories and a concise summary of the numerical information displayed. All labels, number representation (i.e. totals, percentage of whole, percentage of category, etc), headers, and report names are customizable in the report interface.

To change the display, merely drag a new dimension from the dimension bar to one of the axes of the table section of the report. Figure 8.4 shows how the report changes when the Claimancy (CL) dimension is dragged to the horizontal axis.

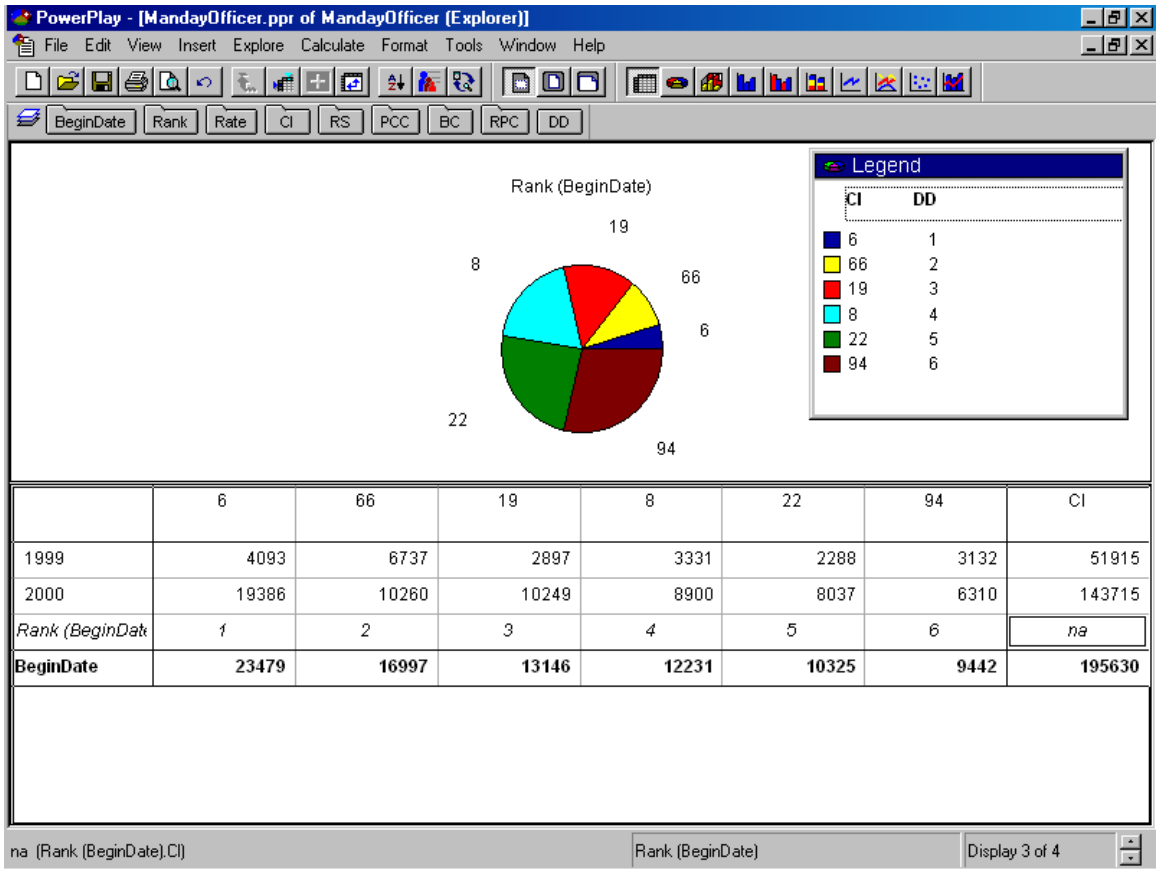


Figure 8.4 Axis Replacements in a Standard Report

After changing the axis dimension, the report now displays the ranked order of Claimancies by total number of manday support provided. The graphical display is also changed to reflect the new axis information. Again, the ranking order and number of elements displayed can be changed to reflect the wishes of the user.

The user may now wish to explore the timing of the support more fully. By double-clicking any of the table element blocks, the underlying information is displayed to the depth allowed by the cube model. In Figure 8.5, the begin date summary field has been selected under Claimancy 66.

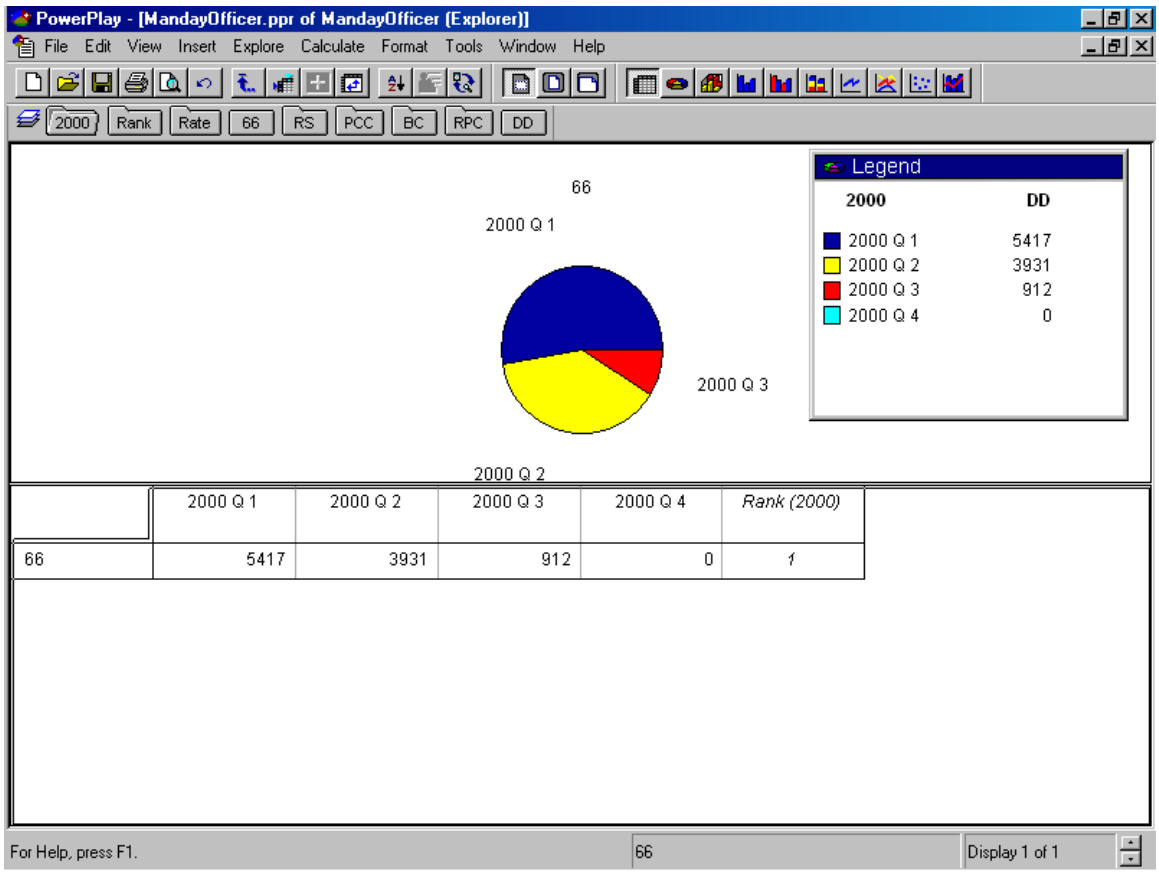


Figure 8.5 Drill-Down Capabilities

The display now reveals the mandays spent on Claimancy 66 by quarter. The graph has also changed to reflect the data. By double-clicking the rank column, we can drill back up to the report shown in Figure 8.4. PowerPlay designates whether you can go up or down by changing the cursor to a plus sign with an up or down carrot in the center if additional information is available for view.

Reports are provided in NaRSDAT to cover mandays expended, manpower available and RPN budget execution. These reports are available in the NaRSDAT Reports folder of the default installation directory. The reports provide information relevant to the metrics described in Chapter 5.

3. PowerPlay Security and Distribution

The data cubes and reports can be distributed throughout a network for user access or may be accessed from a single machine. Either way, some level of security and user preferences must be implemented. PowerPlay implements security through a user

database. Each user is assigned the permission to access the data cubes and reports. The user is also assigned permissions that either enable or disable his ability to change the view of the reports he is able to access.

Data cube security is much the same as that for the report viewer. Users can be given the permission to change existing reports or create new reports for distribution. It is useful to remember that a report is merely a viewing template for the data cube. No changes are made to the data cube in any way by changing the PowerPoint Report or changing the Power Point Explorer settings. However, many elements of an organization's data may not be intended for general consumption. The ability to target the type and depth of data available to each user is both a security requirement and a productivity enhancer. This prevents the user form needlessly exploring data that has no relevance to the user's role in the organization.

B. INITIAL IMPLEMENTATION

NaRSDAT will be used as a proof of concept application for the data warehousing effort in the Naval Reserve Force. NaRSDAT will provide a basis from which a full data warehousing and decision support system can be implemented. The initial funding for an Oracle based data warehouse solution has already been initiated. The author and the NaRSDAT development have provided key feedback for development and requirements generation of the Oracle based solution at CNRF.

A complete user's guide is provided in Appendix D. The user's guide provides information about system requirements to run NaRSDAT and installation instructions. It also provides information concerning the initial population of the data warehouse and requirements for subsequent data updates. Lastly, the user's guide provides web addresses and e-mail addresses for technical support for all areas of the development. The Visual Basic code is supplied in Appendix E along with the SQL queries used in the data cleaning and migration process. A qualified administrator with a moderate level of knowledge regarding Visual Basic and Microsoft Access will be able to install and maintain the prototype using only this information and the online resources provided.

Pending the completion of the Oracle development, NaRSDAT will be used to provide information to all Deputy Chiefs of Staff (DCOS) and to Commander, Naval

Reserve Force (CNRF). The software will be installed on a stand-alone system in the N8 office spaces. The information provided for the DCOSs will be generated and distributed from this central location. Assessment information will be distributed to support policy change evaluation as well as personnel and deployment concerns.

NaRSDAT will also provide data warehouse facilities for the separate decision support tool that has been purchased by CNRF. The formatted data will be accessed by the decision support tool to provide a more targeted ability to forecast policy effects on the force.

There are initial limitations to the deployment. The data warehouse only has end of year data from fiscal year 2000 available. Until the system has been in place for several months, the decision support capabilities will be extremely limited. The initial deployment period will be a test period where existing data sources will be compared to NaRSDAT and any discrepancies will be noted and corrected. During this initial period, both the legacy data channel and the data warehouse channel will be required to ensure proper operation of the force. This testing period will be crucial in overcoming any departmental doubts as to the legitimacy and precision of the information.

IX. CONCLUSIONS AND FUTURE WORK

A. SUMMARY

The Naval Reserve Force identified a significant problem with their assessment process, which was manpower intensive and not structured to provide a timely assessment of the force. The legacy databases that CNRF relied upon for information concerning the results of policy changes were not flexible enough to provide the required feedback for the evaluation process in the strategic decision making process.

We developed a software solution for the assessment process in order to give Commander, Naval Reserve Force a tool that could make the assessment process a strategic part of Naval Reserve leadership. Combining current COTS applications with original code derived from requirements research and data warehouse design, a proof of concept system called NaRSDAT was constructed. NaRSDAT provides CNRF with an OLAP and preliminary data mining tool based on a data warehouse derived from six existing legacy databases.

By leveraging well established COTS application technology and integrating it with a Visual Basic program specifically designed to meet NRF needs, NaRSDAT provides a custom solution to CNRF and a working prototype to establish the usefulness and benefits of a strategic decision support system.

B. NEXT GENERATION TOOL

NaRSDAT has provided the basis for a full scale development in the data warehousing and decision support arena at NRF. The planned development is for an Oracle database and data warehouse design. The OLAP and data mining solution is to employee Oracle Discoverer as the data mining application and Expert Choice as the decision support tool. Both of these tools will access the data warehouse.

Future rollouts are already planned to allow the inclusion of recruiting and retention data, as well as more operational support information. The plan is to develop a complete Naval Reserve Force data warehouse that can provide coverage of forecasting and decision support for all areas of policy and force assessment.

The full scale development will provide superior transaction tracking and backup solutions for the data warehouse. The development is planned to employ a dedicated server box with robust multi-processor capability and a load balancing operating system. This allows for a scalable deployment from the ground up. As information requirements grow, so can the data warehouse. Minimal additional development is required for the expansion.

C. DISTRIBUTED SOLUTIONS

NaRSDAT was not developed to provide a distributed solution. The database and engine are not sufficiently robust to provide the security and transaction features required of a distributed solution. NaRSDAT does provide the local proof of concept that will enable future expansion of the data warehouse concept.

The initial design of the Oracle solution is to provide a centralized solution to the data warehousing requirement. Plans include providing a Local Area Network (LAN) solution in the next rollout and then to turn on the web enabled features of the Oracle package to provide the same assessment information to the force at large. The key advantage to this will be to provide decision support and assessment information to the Naval Reserve Force personnel in Washington, DC that provide support through the requirements and funding process.

Specific reports and graphs will be developed to provide targeted solutions to all levels of command in the Naval Reserve Force. All commands from Echelon VI to Echelon II will have the ability to access information directly related to their performance and to use that information to tailor the performance and policy of the organization from their command level downward.

D. IMPLICATIONS FOR THE FORCE

There are far reaching implications for the Naval Reserve Force in the implementation of a complete, distributed data warehouse and decision support solution. Traditional power lies in the ability to control quality and distribution of data to the benefit of the department or organization. The power base that exists will be very reluctant to relinquish any of this control. Since the data warehouse solution is very

complete, they stand to lose an apparent base of power. A significant effort will be required to ensure the current knowledge brokers that their positions are secure and that they will continue to be required for the proper functioning of the Naval Reserve Force.

Security is another major concern. With the current acceleration of web use, the ability to provide access to all legitimate users and restrict use from hackers and other disreputable Internet sources is of paramount importance. Since relevant personal information such as SSN, name, birth date, and other information concerning deployment dates and resources will be available via a network connection, a significant effort must be undertaken to restrict access to this information. National security and personnel morale are drastically affected by problems in the security arena.

There is also a tendency to attempt to run day-to-day management functions with the data warehouse. This urge must be avoided. This system is designed to be a planning and forecasting tool for long term processes. An attempt to use this information in a management function would decrease morale and cause inaccurate policy decisions. The troops get frustrated when management makes changes based on slight fluctuations in data that may be a seasonal or environmental fluctuation. The users must understand what they are being shown and how it applies to their job. Placing the reports generated by the data warehouse into context for the user is a primary requirement of the developer.

E. FUTURE RESEARCH OPPORTUNITIES

There are several possibilities for follow on research with relation to the data warehouse development at CNRF. Research can be conducted specifically for any of the future rollouts that will be attempted. An especially interesting topic would be to evaluate the full implications of web enabled deployment of the data warehouse and decision support system. Research about a specific database and its effects on the data warehouse would also be a possibility. Selecting a particular legacy database, the researcher could evaluate the required dimensional models and then determine how the available data might influence the decision process.

Research into the results of the decision support portion of the data warehouse deployment would be very beneficial to CNRF. Following several decisions from

information discovery and policy change through data warehouse results would indicate the level of benefit to CNRF of the decision support tool and suggest possible ways to make better decisions with the information at hand. Traditionally, policy changes are directed towards a specific department or phase of operations. That department or phase is then closely watched to see if the desired change took place. In most instances, the change actually effected more than the single department. These peripheral changes can take many evaluation cycles to find because there is no direct indication of the cross effect of the policy change. With NaRSDAT and the follow on implementation, these direct and indirect effects can easily be monitored. All relevant impact of the policy change will be evident due to the relational nature of the data warehouse.

As a last area, research could be conducted supporting the possible use of Artificial Intelligence (AI) systems in conjunction with the data warehouse system. One very interesting area of study with regards to AI would be to evaluate the benefits of an AI system to supplement personnel in key positions that rotate on a frequent basis. If a unit Commanding Officer, who is rotated every 18 months, had an AI asset programmed with normal responses for typical command decisions, he would have a valuable tool for consistent leadership and policy decisions.

The NaRSDAT prototype developed in this document and the full Oracle implementation will provide Commander, Naval Reserve Force with a flexible, directed environment to assist in the strategic decision making process. These tools will provide CNRF with a core of relevant metrics by which the force can be assessed as well as a way to maintain those metrics without bias. Most significantly, it will enable NRF to intelligently and accurately manage the decision process and provide relevant feedback to its internal and external customers. In today's recruiting and operational environment, this could mean the difference between success and irrelevance for sustaining the Naval Reserve Force.

APPENDIX A. LEGACY METADATA

A. GLOBAL METADATA

Data Element Name	Position	# of characters	Type of entry	Description
NRA	1	4	number	Naval Reserve Activity number
RUIC	2	5	alpha-num	Reserve Unit Identification Code
RBSC	3	4	number	Reserve Billet Sequence Code number
ECPC	4	1	alpha-num	Expanded Compensation Pay Code
CHG	5	1	alpha	Billet Change Indicator
UMUIC	6	5	number	Unit Mobilization UIC number
BAUIC	7	5	alpha-num	Billet Active duty UIC
ARATE	8	5	alpha-num	Rating/designator for billet
V	9	1	alpha-num	Vertical rank substitution code
HZ	10	2	alpha	Horizontal rate/designator substitution code
SX	11	1	alpha	Sex code for billet (m/f/e)
PNEC	12	4	number	Primary NEC/NOBC of billet
SNEC	13	4	number	Secondary NEC/NOBC of billet
BENDT	14	8	number	Billet end date
ADTE	15	8	number	Date assigned to the billet
IRATE	16	5	alpha-num	Member's rating/designator
ISX	17	1	alpha-num	Member's sex code (- = female)
SSN	18	9	number	SSN of member in billet
MOB	19	3	alpha	Mobilization status code
Name	20	4	alpha	First 4 letters of last name of member filling billet
IRAD	21	2	alpha-num	Readiness code of member in billet
Q	22	1	alpha-num	Next training element needed
M	23	1	alpha-num	Month IRAD last updated
Y	24	1	alpha-num	Year IRAD last updated
NECP	25	4	number	Member's Primary NEC/NOBC
NECS	26	4	number	Member's Secondary NEC/NOBC
TRUIC	27	5	alpha-num	Training UIC
ABSC	28	5	alpha-num	Active Billet Sequence Code
Unit	29	10	alpha-num	Unit long name
RPC	30	3	number	Reserve Program Code
BILTITLE	31	23	alpha-num	Billet title
OE	32	1	alpha	Officer Enlisted flag (o/e)
REC	33	1	alpha	Record type (Local, iaP, caI, caO, Vacant)
Groups	34	15	alpha-num	Organizational relationships
PEBD	35	8	number	Pay Entry Base Date
DOR	36	8	number	Date of Rate
DOB	37	8	number	Date of Birth

DTASG	38	8	number	Date assigned to unit
IAP	39	1	alpha	IAP indicator (V)
IAPDATE	40	8	number	Date assigned to IAP status
PGRADE	41	3	alpha-num	Member's paygrade
BGRADE	42	3	alpha-num	Billet paygrade
CMD	43	2	number	Command number
PGCD	44	2	alpha-num	Activity Program Code
AUT	45	3	number	Mobilization activity code
ACMD	46	2	number	Mobilization activity APC command code
APG	47	2	alpha-num	Activity Program Code
AUC	48	3	number	Mobilization activity code
AMGR	49	6	alpha-num	Air program manager code
SMGR	50	6	alpha-num	Surface program manager code
PRI	51	2	alpha	Priority unit indicator
BPRI	52	2	alpha	Billet priority indicator
A	53	1	alpha	Billet structuring alignment flag
I	54	1	alpha	Billet structuring info flag
D	55	1	number	Drill pay code (1 = pay, 9 = nonpay)
PCMD	56	2	number	Member's command
PNRA	57	4	number	Member's NRA
BIN	58	7	alpha-num	Billet Identification Number
City	59	18	alpha	Member's city
St	60	2	alpha	Member's state
ZIP	61	5	number	Member's zipcode

Table A.1 Global Legacy Database Metadata

B. RTSS METADATA

Data Element Name	Position	# of characters	Type of entry	Description
RUB_NRA_CMD	1	4	number	Naval Reserve Activity number
SORT_RU_UIC	2	5	alpha-num	Reserve Unit Identification Code
B_RBSC	3	4	number	Reserve Billet Sequence Code number
B_NBR_DRL_CD	4	1	alpha-num	Expanded Compensation Pay Code
A,*,d	5	1	number	Billet Change Indicator
B_AUIC	6	5	number	Unit Mobilization UIC number
RUB_AUIC	7	5	alpha-num	Billet Active duty UIC
B_RTNG_DESG_CD	8	5	alpha-num	Rating/designator for billet
B_FAC_VERT	9	1	alpha-num	Vertical rank substitution code
B_FAC HORIZ	10	2	alpha	Horizontal rate/designator substitution code
B_BIL_SEX_CD	11	1	alpha	Sex code for billet (m/f/e)
B_PRI_NEC_NOBC	12	4	number	Primary NEC/NOBC of billet
B_SEC_NEC_NOBC	13	4	number	Secondary NEC/NOBC of billet

B_BLT_END_DT	14	8	number	Billet end date
ADTE	15	8	number	Date assigned to the billet
P_RTN_DESG	16	5	alpha-num	Member's rating/designator
P_SEX_CD	17	1	alpha-num	Member's sex code (- = female)
P_SSN	18	9	number	SSN of member in billet
P_MOB_STAT_CD	19	3	alpha	Mobilization status code
P_NAME	20	4	alpha	First 4 letters of last name of member filling billet
PB_IRAD_PCT_CD	21	2	alpha-num	Readiness code of member in billet
PB_QUAL_CD	22	1	alpha-num	Next training element needed
PB_IRAD_MN	23	1	alpha-num	Month IRAD last updated
PB_IRAT_YR	24	1	alpha-num	Year IRAD last updated
P1_NEC_NOBC	25	4	number	Member's Primary NEC/NOBC
P2_NEC_NOBC	26	4	number	Member's Secondary NEC/NOBC
RUP_UIC	27	5	alpha-num	Training UIC
B_ABSC	28	5	alpha-num	Active Billet Sequence Code
AU_LONG_TITL	29	10	alpha-num	Unit long name
PGM_RPC_NO_CD	30	3	number	Reserve Program Code
B_BIL_DN	31	23	alpha-num	Billet title
SORT_OFF_ENL_CD	32	1	alpha	Officer Enlisted flag (o/e)
REC_TYP	33	1	alpha	Record type (Local, iaP, caI, caO, Vacant)
RUB_SELECT_CD	34	15	alpha-num	Organizational relationships
P_PEBD	35	8	number	Pay Entry Base Date
P_DOR	36	8	number	Date of Rate
P_DOB	37	8	number	Date of Birth
PA_DT_ASG	38	8	number	Date assigned to unit
P_IAP_FLAG	39	1	alpha	IAP indicator (V)
P_IAP_DATE	40	8	number	Date assigned to IAP status
PGP_PAYGD	41	3	alpha-num	Member's paygrade
PGB_PAYGD	42	3	alpha-num	Billet paygrade
RUB_ACT_CMD_N	43	2	number	Command number
RUB_ACT_PG_CD	44	2	alpha-num	Activity Program Code
RUB_ACT_UNT_CD	45	3	number	Mobilization activity code
RUP_APC_CMD_NO	46	2	number	Mobilization activity APC command code
RUP_	47	2	alpha-num	Activity Program Code
RUP_ACT_UNT_CD	48	3	number	Mobilization activity code
AU_AIR_PGM_MGR	49	6	alpha-num	Air program manager code
AU_SURF_PGM_M	50	6	alpha-num	Surface program manager code
RUP_PRI_TYCD	51	2	alpha	Priority unit indicator
RUB_PRI_TYCD	52	2	alpha	Billet priority indicator
BS_ALIGN_FLG	53	1	alpha	Billet structuring alignment flag
BS_INFO_FLG	54	1	alpha	Billet structuring info flag
PA_DRL_PAY_CD	55	1	number	Drill pay code (1 = pay, 9 = nonpay)
RUP_CMD_NO	56	2	number	Member's command
RUP_NRA_CD	57	4	number	Member's NRA
B_BLT_ID	58	7	general	Billet Identification Number
PAD_CITY	59	18	alpha	Member's city

PAD_STATE	60	2	alpha	Member's state
PAD_ZIP	61	5	number	Member's zipcode

Table A.2 RTSS Legacy Database Metadata

C. MANDAY METADATA

Data Element Name	Position	# of characters	Type of entry	Description
PMC	1	4	alpha-num	Program Manager Code
ISSACT	2	17	alpha	Issuing activity of orders
RUIC	3	6	alpha-num	Reserve Unit Identification Code of unit of member on orders
Unit	4	18	alpha-num	Short name of unit of member on orders
RPTDATE	5	10	alpha-num	Reporting date of orders
SDN	6	15	alpha-num	Standard document number of orders
Name	7	5	alpha	First 5 letters of last name of member on orders
Rate	8	5	alpha-num	Rate of member on orders
SSN	9	9	number	SSN of member on orders
RPTUIC	10	6	alpha-num	Unit Idntification Code of reporting activity on orders
RPTCMD	11	18	alpha-num	Short name of reporting activity on orders
DUTYDAYS	12	3	number	Number of days of active duty
TRVLDAYS	13	1	number	Number of days of travel
TRVLCODE	14	1	number	Code for type of travel on orders
BUDCAT	15	2	alpha	Budget category for expenditures on orders
RPC	16	2	number	Reserve Program Code of orders
OE	17	1	alpha	Officer or enlisted flag (o/e)
TRYDATE	18	10	number	Date orders entered into system
APPRDATE	19	10	number	Date orders approved
READYPRINT	20	10	number	Date orders ready to print
PRINTDATE	21	10	number	Date orders printed
PGRADE	22	3	alpha-num	Paygrade of member on orders
NUMODS	23	2	alpha	Numbers of modifications to orders to date
RENTALCAR	24	1	alpha	Rental car authorization flag (y/n)
TRACKCD	25	9	alpha-num	Tracking code
CANAX	26	1	alpha	not required. Indicator for reason of cancelled orders.
CANAXDATE	27	6	number	If CANAX, date of cancellation
CIN	28	10	number	Not required. Course Identification Number if orders are for school.

Table A.3 Manday Legacy Database Metadata

D. WINPAT METADATA

Data Element Name	Position	# of characters	Type of entry	Description

UIC	1	5	alpha-num	Unit identification code
CL	2	2	number	Claimancy code
RS	3	2	number	Resource sponsor
UIC_TITLE	4	40	alpha-num	UIC Long name
PE	5	8	alpha-num	Unknown
DS	6	6	alpha-num	Unknown
DS$	7	4	number	Money spent on claimancy
E/S	8	3	number	End strength

Table A.4 WinPAT Legacy Database Metadata

E. RIMS(FM) METADATA

Data Element Name	Position	# of characters	Type of entry	Description
Fiscal Year	1	4	number	Fiscal year of orders
Document type	2	2	alpha	Type of orders
SDN	3	17	alpha-num	Standard document number of orders
Officer/Enlisted code	4	1	alpha	O for Officer, E for Enlisted
ACRN	5	2	alpha	
Begin Date	6	9	alpha-num	Date orders begin
Number of days	7	3	number	Number of active duty days
Travel Days	8	1	number	Number of travel days authorized
Initiation amount SUM	9	8	number	Prepaid amount of orders
Accounts payable amount SUM	10	8	number	Amount claimed but not billed
Expenditure amount SUM	11	8	number	Amount payed on orders
Program Code	12	2	number	Program Code for orders
Program category code	13	2	number	Program category code for orders
SSN	14	9	number	SSN for individual on orders

Table A.5 RIMS(FM) Legacy Database Metadata

F. TFFMS METADATA

Data Element Name	Position	# of characters	Type of entry	Description
ANAME_A	1	30	alpha	Reserve unit long name
SHORT_NAME	2	16	alpha	Reserve unit short name
ACODE	3	10	number	
CLMT_NAME	4	10	alpha	Reserve unit claimancy name
CLMT_CODE_	5	2	number	Reserve unit claimancy code
SMC_A	6	2	alpha-num	
LOCATION	7	18	alpha	City, state, country
PREDOM_RS	8	3	alpha-num	Resource Sponsor

ATYPE	9	1	alpha	
AODC	10	2	alpha-num	
PRED_AGSAG	11	4	alpha-num	
SUI	12	1	number	
SEA_SHORE	13	1	number	1 - sea, 2 – shore
MCA	14	1	alpha	
PKT_NUMBER	15	6	number	
PKT_DATE	16	7	number	
DWNLD_DATE	17	7	number	
GEO_LOC	18	8	alpha-num	Geographic location code
MARP	19	4	number	
PREDOM_PE	20	8	alpha-num	
ANAME_B	21	30	alpha	Active unit long name
AUIC	22	5	number	Active unit identification code
SMC_B	23	2	alpha-num	
CLMT_CODE2	24	2	number	Billet claimancy code
BIN	25	7	number	Billet identification number
BSC	26	5	number	Billet sequence code
TITLE	27	40	alpha-num	Billet title
EFF_BGN_DT	28	5	number	Billet begin date (unused)
EFF_END_DT	29	5	number	Billet end date (unused)
RI	30	1	alpha	
MEC	31	1	alpha	
MRC	32	2	alpha	
RS	33	3	alpha-num	Resource sponsor
AGSAG	34	4	alpha-num	
PE	35	8	alpha-num	
PFAC	36	1	alpha-num	
SFAC	37	1	alpha-num	
R_DESIG	38	4	number	Reserve designator
R_GRADE	39	1	alpha	Reserve rank code
R_PNOBC	40	4	number	Reserve Primary Officer Billet Code
R_SNOBC	41	4	number	Reserve Secondary Officer Billet Code
R_PAQD	42	3	alpha-num	Primary AQD
R_PSUB	43	5	alpha-num	Reserve primary subspecialty code
R_SSUB	44	5	alpha-num	Reserve secondary subspecialty code
A_DESIG	45	4	number	Active designator
A_GRADE	46	1	alpha	Active rank code
A_PSUB	47	5	alpha-num	Active primary subspecialty code
A_SSUB	48	5	alpha-num	Active secondary subspecialty code
A_SAQD	49	3	alpha-num	Secondary AQD
R_RTABBR	50	5	alpha-num	Reserve rate abbreviation
R_PNEC	51	4	number	Reserve primary NEC
R_SNEC	52	4	number	Reserve secondary NEC
A_RTABBR	53	5	alpha-num	Active rate abbreviation
A_PNEC	54	4	number	Active primary NEC

A_SNEC	55	4	number	Active secondary NEC
R_PAYGRADE	56			Reserve paygrade
R_OCC_SRS	57			
R_PAYPLN	58			
A_PAYGRADE	59			Active paygrade
A_OCC_SRS	60			
A_PAYPLN	61			
A_PS	62			
A_HS	63			
RFC	64	3	alpha-num	
LANG_ID	65	2	alpha	Language ID code
LISTEN	66	2	number	Percent fluent
READ	67	2	number	Percent fluent
SPEAK	68	2	number	Percent fluent
WRITE	69	2	number	Percent fluent
MPWR_CAT	70	1	alpha	Manpower category (E/O/C)
MPWR_SHORT	71	3	alpha	Manpower short name (ENL/OFF/CIV)
AUTH_MPWR	72	1	alpha	Authorized manpower category
MOB_BGN	73	2	number	
MOB_END	74	2	number	
PR	75	1	number	
PREV_UIC	76	5	number	Previous billet UIC
PREV_BSC	77	5	number	Previous billet sequence code
ALT_MPWR	78	1	alpha	Alternate manpower code
ALT_CAT	79	1	alpha	Alternate manpower category
ALT_SHORT	80	3	alpha	Alternate manpower short name
PRI_CAT	81	1	alpha	Reserve manpower category
PRI_MPWR	82	1	alpha	Reserve manpower code
PRI_SHORT	83	3	alpha	Reserve manpower short name
RUIC	84	5	alpha-num	Reserve unit identification code
APPROP_CAT	85	3	alpha	Appropriation category
R_E_PAYGRD	86			
A_E_PAYGRD	87			
BIN_PKT	88	6	number	Bin packet number
BIN_DATE	89	7	number	BIN date
RECORD_TYP	90			
HAIC	91			
DMC	92	5	number	
EMC	93	4	alpha-num	
ADDU_BIN	94	7	number	
ADDU_FITRP	95	1	number	
MPWR_AC	96	1	alpha	

Table A.6 TFMMS Legacy Database Metadata

THIS PAGE INTENTIONALLY LEFT BLANK

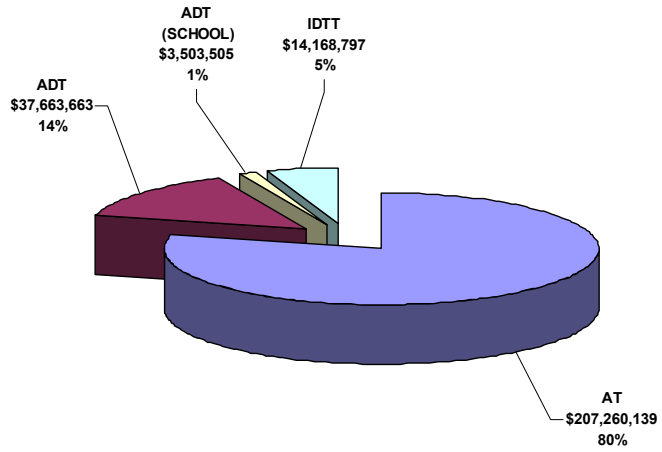
APPENDIX B. SAMPLE REPORTS

A. RPN EXECUTION SAMPLE REPORTS

RPN Execution

The second annual Shareholders' Report begins with the process of historical views. If last year's report was a snapshot of the year, the FY-00 begins a process of establishing a moving picture of two years of data/information. With next year's FY-01 report, three years of data will help to complete the picture. Not only will irregularities in the data be high-lighted within a fiscal year but trends will be examined over a three year timeframe.

FY-00 OVERALL RPN EXECUTION



FY-00 Review

A review of AT, ADT, ADT (Schools) and IDTT shows an execution of $262.6M. The crossover/ reimbursables/no-cost orders of $12.8M and the retirement account of $35M are not graphed. See page 54 for FY-00 spreadsheet.

FY-99/00 Comparison

-In FY-00 RPN execution decreased by $14.9M (5%).

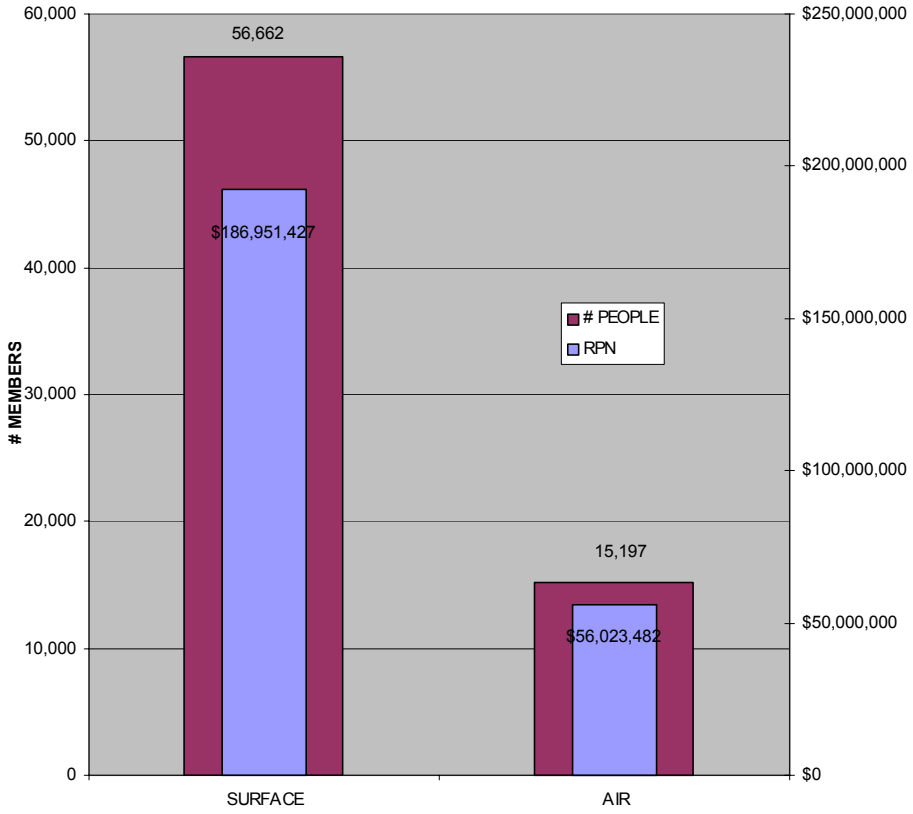
-ADT (Schools) expenditure declined by $1.9M (35%).

-AT, ADT, and IDTT remained proportionally the same in both years.

8

Figure B.1 Overall RPN Sample Report

Air/Surface RPN Split



FY-00 Review

-79% of the members were Surface.

-21% of the members were Air.

-Of an expense of $248M, Surface was 77% of the cost, and Air was 23%.

FY-99/00 Comparison

-Cost percentages were level across years. Member percentages showed a minor variance.

9

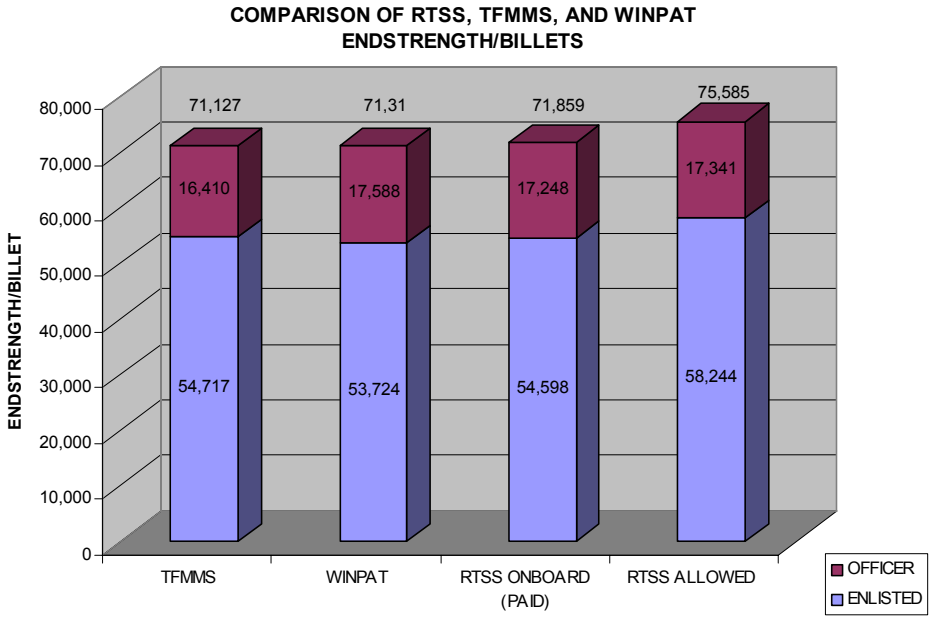
Figure B.2 Air Surface RPN Sample Report

B. MANPOWER/PERSONNEL SAMPLE REPORTS

Manpower FY-00

In the Naval Reserve people are our most important asset. For FY-00, End Strength (RTSS/Global), Requirements (TFFMS), Budgeting (WINPAT) and Utilization (RIMS/MANDAYS) were reviewed for the capture of Manpower.

End Strength (Billets) FY-00



FY-00 Review

-TFMMS and WINPAT, and RTSS Onboard are closely aligned for both officers and enlisted.

-RTSS Allowed numbers are higher than WINPAT, TFMMS, and RTSS Onboard.

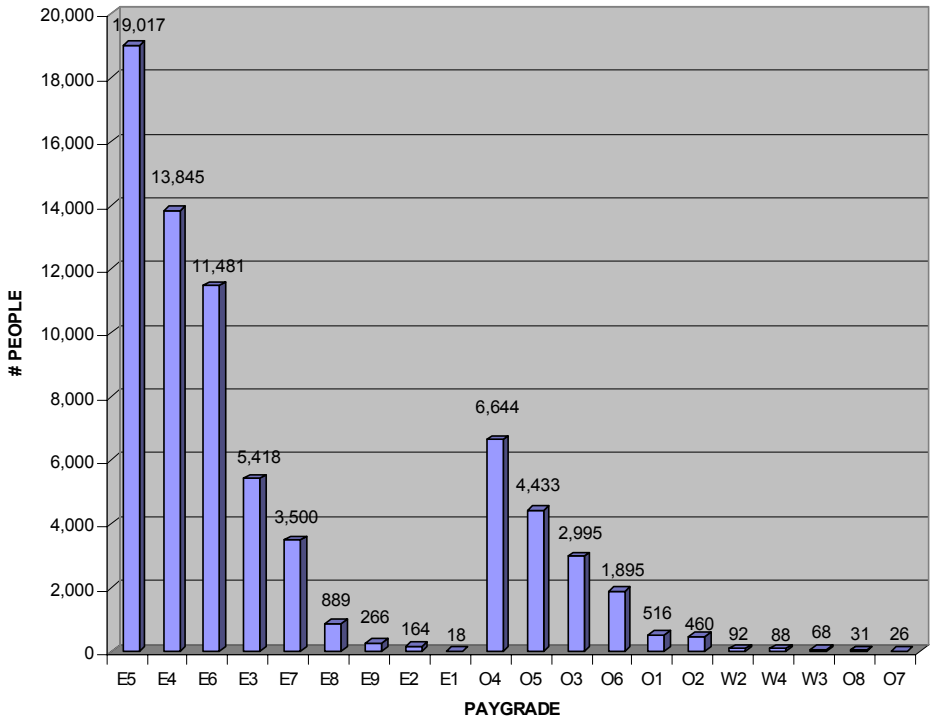
-RTSS Allowed (enlisted) is 6% above TFMMS and 8% above WINPAT.

FY-99/00 Comparison

-A FY-00 TFMMS realignment explains the negative 29% difference when compared to FY-00.

Figure B.3 FY Manpower Sample Report

Personnel by Pay Grade FY-00



FY-00 Review

-The Total Paid Force at the end of FY-00 was 71,859 people.

-The Officer count was 17,194.

-The Enlisted count was 54,352.

-The official RCCPDS numbers were 17,194 officers and 54,352 enlisted for a Total Force of 71,546.

FY-99/00 Cross Comparison

-FY-00 and FY-99 Global paid members maintained close pay grade percentages across years.

-FY-00 and FY-99 TFMMS maintain close pay grade percentages across years.

19

Figure B.4 Manpower By Rate Sample Report

C. MANDAY SAMPLE REPORTS

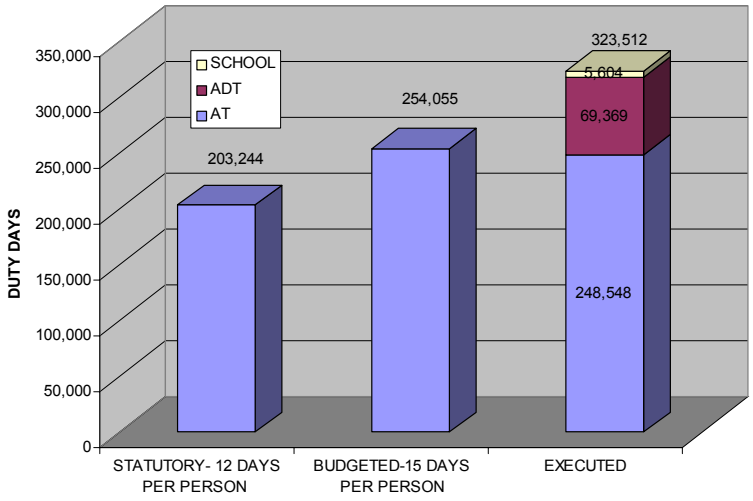
TYPE OF SUPPORT

This section reviews the Naval Reserve from a "Customer" prospective. The Naval Reserve provides trained personnel to our customer (Claimants) by blocks of Man Days, specifically a "Duty Day". A Duty Day is one Man Day of Support to the customer not including Travel Days. Travel Days are accounted for separately since they do not add value to the customer. The three categories covered in this section include: Annual Training (AT); Active Duty for Training (ADT); and Active Duty for Training Schools (ADT Schools). Further analysis will show the Naval Reserve Program Manager perspective.

Officer Statutory Vs Executed

Year-end end strength for officers was adjusted to a 99% level of statutory performance. The standard for statutory compliance was based on a 12 Duty Day requirement for AT/ADT/ADT (Schools).

OFFICER STATUTORY & BUDGETED VS. EXECUTED



FY-00 Review

Executed days exceeded the 12 statutory day requirement by 120,277 Duty Days (59%).

Executed days exceeded the budgeted 15 days by 69,466 days (27%).

FY-99/00 Comparison

-Plus-ups in funding impacted Duty Day usage in FY-99.

-In FY-99, Executed Duty Days exceeded 12 day statutory requirements by 126,687 Duty Days (61%). While in FY-00, executed days exceeded statutory requirements by 120,277 Duty Days (59%). The profile of execution is relatively constant across years.

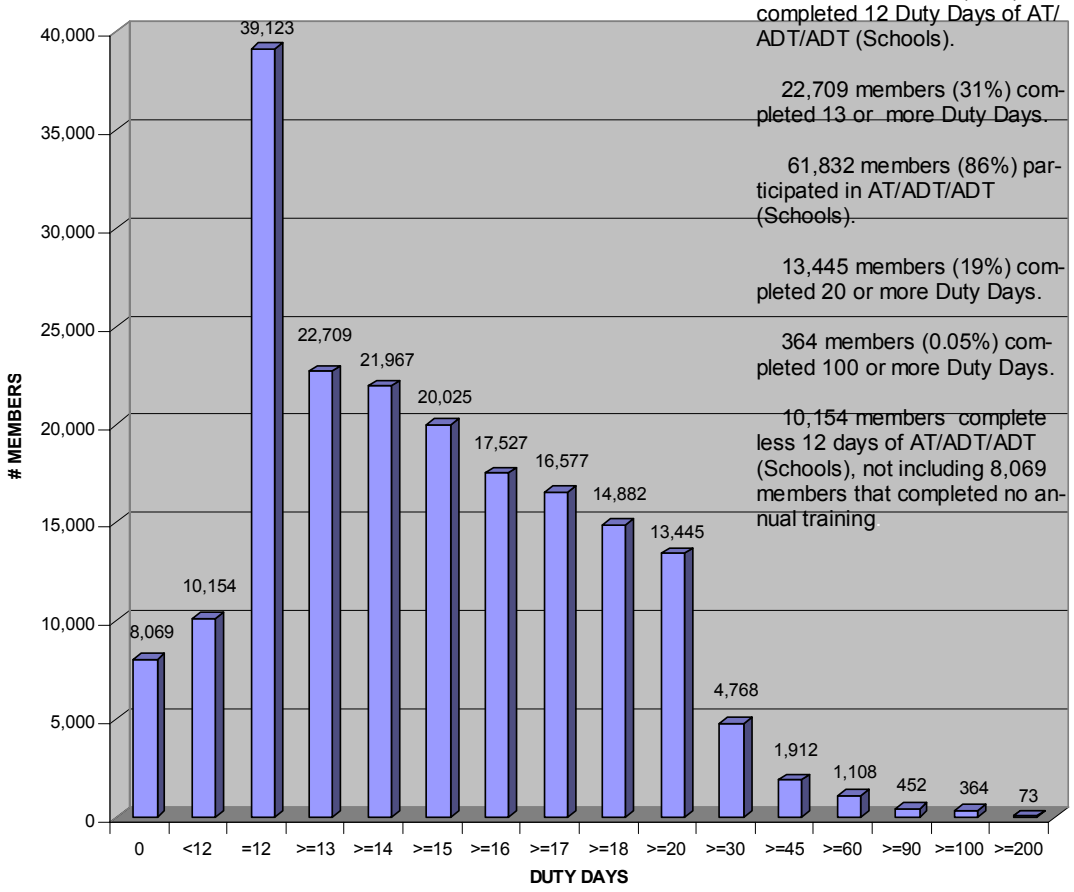
-The executed Duty Day figures do not take into account those members that exceed requirements, or those members not complying.

28

Figure B.5 AT Execution Sample Report

Members Completing AT/ADT/ADT (Schools) by Duty Days

FY-00 Review



FY-99/00 Comparison

-End strength declined by 2%.

-Duty Day usage increased by 2%.

-Members with over 20 Duty Days increased by 37%.

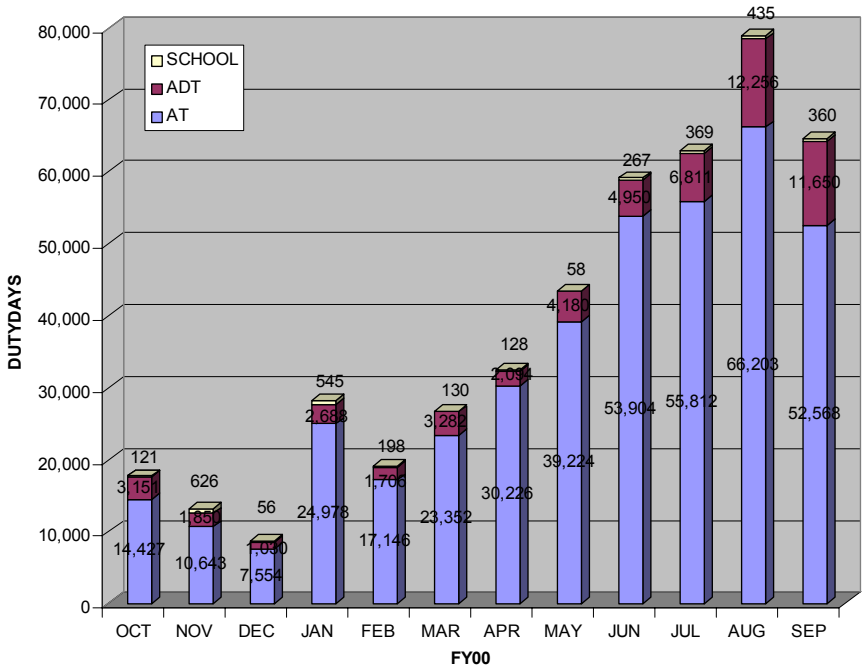
-Members with less than 12 Duty Days declined by 49%.

32

Figure B.6 Execution By Number Of Days Sample Report

D. SUPPORT SAMPLE REPORTS

Exercise AT/ADT/ADT (Schools)



FY-00 Review

-459,280 Duty Days were expended for Exercises. 401,248 (87%) were AT; 54,862 (12%) were ADT; and 3,170 (1%) were ADT Schools.

-4th Quarter was the high quarter for AT, ADT and ADT (Schools).

-August was the high month for AT and ADT consumption; November was the high month for ADT (Schools) consumption.

FY-99/00 Comparison

-Data was not available for the 1st Qtr. of FY-99.

-Quarterly expense patterns are not proportional for all quarters across years.

-AT Duty Days expended appear to have declined in FY-00.

-ADT Duty Days expended appear to have declined in FY-00.

-Quarterly expense patterns are proportional for the 2nd, 3rd and 4th Quarter across years.
4th Qtr. remains the highest Qtr. for ADT (Schools) but shows a greater reduction in Duty Days than the 2nd and 3rd Qtrs.

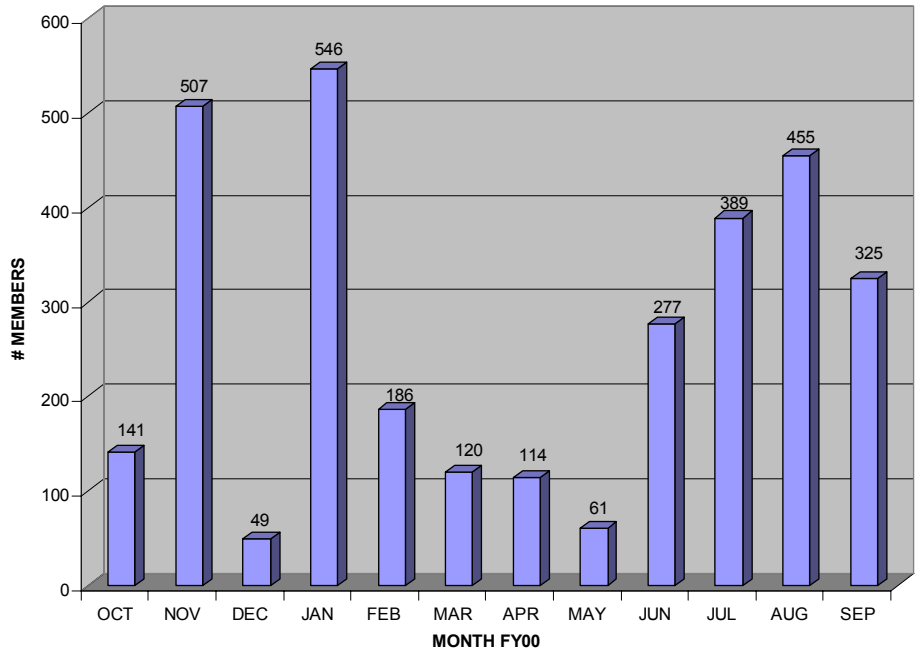
-ADT (Schools) Duty Days expended appear to have declined in FY-00.

43

Figure B.7 Exercise School Sample Report

ADT (Schools) Exercises

.



FY-00 Review

-The high month was January (546 days). The low month was December (49 days).

-The high quarter was the 4th Quarter (1,169 days). The low quarter was the 3rd Quarter (452 days).

-The Quarterly Average was 792 Duty Days. The 1st and 2nd quarters were closest to the yearly mean.

FY-99/00 Comparison

-Duty Days dropped dramatically between FY-99 and FY-00, even without calculating FY-99 1st Quarter usage in the equation.

-The percent of total for the months could not be statistically compared due to three months of unavailable data. However, it appears that the usage pattern by month is not consistent over time, or between years.

Figure B.8 ADT Scools For Exercise Sample Report

APPENDIX C. RELATIONAL TABLE METADATA

A. TABLE: ACTIVEUNIT

Columns

Name		Type	Size
AUIC		Text	5
	AllowZeroLength:	False	
	Required:	True	
	Source Field:	Global.ACMD	
CL		Text	50
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.CMD	

Relationships

ActiveUnitBillet

ActiveUnit		Billet
AUIC	1	AUIC(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

B. TABLE: BILLET

Columns

Name		Type	Size
BilletIDNumber		Text	7
	AllowZeroLength:	False	
	Required:	True	
	Source Field:	Global.BIN	
NRA		Text	5
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.NRA	
RUIC(fk)		Text	5

AllowZeroLength: False
Required: True
Source Field: ReserveUnit.RUIC

AUIC(fk) Text 5
AllowZeroLength: False
Required: True
Source Field: ActiveUnit.AUIC

BilletRate Text 5
AllowZeroLength: False
Required: False
Source Field: Global.Arate

BilletGrade Text 3
AllowZeroLength: False
Required: False
Source Field: Global.BGRADE

PrimaryNEC Text 4
AllowZeroLength: False
Required: False
Source Field: Global.PNEC

SecondaryNEC Text 4
AllowZeroLength: False
Required: False
Source Field: Global.SNEC

ReservePgmCode Text 3
AllowZeroLength: False
Required: False
Source Field: Global.RPC

VerticalSubCode Text 1
AllowZeroLength: False
Required: False
Source Field: Global.V

HorizSubCode Text 2
AllowZeroLength: False

Required: False
Source Field: Global.HZ

SexCode Text 1
AllowZeroLength: False
Required: False
Source Field: Global.SX

BilletName Text 50
AllowZeroLength: False
Required: False
Source Field: Global.BILTITLE

UnitName Text 30
AllowZeroLength: False
Required: False
Source Field: Global.Unit

Relationships

ActiveUnitBillet

ActiveUnit		Billet
AUIC	1	AUIC(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

BilletMemberInBillet

Billet		MemberInBillet
BilletIDNumber	1	BilletIDNumber(fk)

Attributes: Enforced, Left Join
RelationshipType: One-To-Many

ReserveUnitBillet

ReserveUnit		Billet
Ruic	1	RUIC(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join

RelationshipType: One-To-Many

C. TABLE: MEMBER

Columns

Name	Type	Size
SSN	Text	9
AllowZeroLength: False		
Required: True		
Source Field: Global.SSN		
Name	Text	4
AllowZeroLength: False		
Required: False		
Source Field: Global.Name		
IRate	Text	5
AllowZeroLength: False		
Required: False		
Source Field: Global.Irate		
PGrade	Text	2
AllowZeroLength: False		
Required: False		
Source Field: Global.PGRADE		
DateOfBirth	Text	8
AllowZeroLength: False		
Required: False		
Source Field: Global.DOB		
DateOfRate	Text	8
AllowZeroLength: False		
Required: False		
Source Field: Global.DOR		
PrimaryNEC	Text	5
AllowZeroLength: False		
Required: False		
Source Field: Global.NECP		

SecondaryNEC Text 5
AllowZeroLength: False
Required: False
Source Field: SecondaryNEC

OEFlag Text 1
AllowZeroLength: False
Required: False
Source Field: Global.OE

SexCode Text 1
AllowZeroLength: False
Required: False
Source Field: Global.SX

IAPStatus Text 1
AllowZeroLength: False
Required: False
Source Field: Global.IAP

IAPDate Text 8
AllowZeroLength: False
Required: False
Source Field: IAPDate

PayEntryBaseDate Text 8
AllowZeroLength: False
Required: False
Source Field: Global.PEBD

City Text 50
AllowZeroLength: False
Required: False
Source Field: Global.City

State Text 2
AllowZeroLength: False
Required: False
Source Field: Global.St

ZipCode Long Integer 4

AllowZeroLength: False
Required: False
Source Field: Global.ZIP

Relationships

MemberMemberInBillet

Member		MemberInBillet
SSN	1	SSN(fk)

Attributes: Enforced, Left Join
RelationshipType: One-To-Many

MemberOrders

Member		Orders
SSN	1	SSN(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

D. TABLE: MEMBERINBILLET

Columns

Name	Type	Size
SSN(fk)	Text	9
AllowZeroLength: False		
Required: True		
Source Field: Member.SSN		
BilletIDNumber(fk)	Text	7
AllowZeroLength: False		
Required: True		
Source Field: Billet.BIN		
DateAssigned	Text	8
AllowZeroLength: False		
Required: True		
Source Field: Global.ADTE		

IRAD		Text	2
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.IRAD	
NextTrainingElement		Text	1
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.Q	
MonthIRADUpdate		Text	1
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.M	
YearIRADUpdate		Text	1
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.Y	
MobStatus		Text	3
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	Global.MOB	
DrillPayCode		Text	1
	AllowZeroLength:	False	
	Required:	False	
	Source Field:	DrillPayCode	

Relationships

BilletMemberInBillet

Billet		**MemberInBillet**
BilletIDNumber	1	BilletIDNumber(fk)

Attributes:	Enforced, Left Join
RelationshipType:	One-To-Many

MemberMemberInBillet

Member		MemberInBillet
SSN	1	SSN(fk)

Attributes: Enforced, Left Join
RelationshipType: One-To-Many

E. TABLE: ORDERS

Columns

Name	Type	Size
SDN	Text	17
AllowZeroLength:	True	
Required:	False	
Source Field:	Manday.SDN	
ReportDate	Date/Time	8
AllowZeroLength:	False	
Required:	False	
Source Field:	Manday.RPTDATE	
DocType	Text	2
AllowZeroLength:	False	
Required:	False	
Source Field:	RIMSFM.DocumentType	
DutyDays	Integer	2
AllowZeroLength:	False	
Required:	False	
Source Field:	Manday.DUTYDAYS	
TravelDays	Integer	2
AllowZeroLength:	False	
Required:	False	
Source Field:	Manday.TRVLDAYS	
AccountPayableSum	Currency	8
AllowZeroLength:	False	
Required:	False	
Source Field:	RIMSFM.Accounts Payable Amount SUM	
ExpenditureSum	Currency	8

AllowZeroLength: False
Required: False
Source Field: RIMSFM.Expenditure Amount SUM

SSN(fk) Text 9
AllowZeroLength: False
Required: False
Source Field: Member.SSN

PgmCatCode Text 2
AllowZeroLength: False
Required: False
Source Field: PgmCatCode

PgmMgrCode(fk) Text 4
AllowZeroLength: False
Required: False
Source Field: ProgCatCode.ProgCatCode

RUIC(fk) Text 6
AllowZeroLength: False
Required: False
Source Field: Manday.RUIC

ReportUIC Text 6
AllowZeroLength: False
Required: False
Source Field: Manday.RPTCMD

TravelCode Text 50
AllowZeroLength: False
Required: False
Source Field: Manday.TRVLCODE

BudgetCat Text 2
AllowZeroLength: False
Required: False
Source Field: Manday.BUDCAT

ReservePgmCode Integer 2
AllowZeroLength: False

Required: False
Source Field: Manday.RPC

NumberOfMods Integer 2
AllowZeroLength: False
Required: False
Source Field: Manday.NUMODS

RentalCar Text 1
AllowZeroLength: False
Required: False
Source Field: Manday.RENTALCAR

Cancelled Text 1
AllowZeroLength: False
Required: False
Source Field: Manday.CANAX

CourseIDNumber Text 15
AllowZeroLength: False
Required: False
Source Field: Manday.CIN

Relationships

MemberOrders

Member		Orders
SSN	1	SSN(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

ProgCatCodeOrders

ProgCatCode		Orders
Program Category Code	1	PgmCatCode

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

F. TABLE: RESERVEUNIT

Columns

Name		Type	Size
Ruic		Text	5
AllowZeroLength:	False		
Required:	True		
Source Field:	Global.RUIC		
RuicName		Text	50
AllowZeroLength:	False		
Required:	False		
Source Field:	Global.Unit		
Claimant		Integer	2
AllowZeroLength:	False		
Required:	False		
Source Field:	WinPAT.CL		
ResourceSponsor		Integer	2
AllowZeroLength:	False		
Required:	False		
Source Field:	WinPAT.RS		

Relationships

ReserveUnitBillet

ReserveUnit		Billet
Ruic	1	RUIC(fk)

Attributes: Enforced, Cascade Updates, Cascade Deletes, Left Join
RelationshipType: One-To-Many

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX D. USER'S MANUAL

A. INSTALLATION

1. System Requirements

Minimum:

80586 processor

64 MB Ram

20 MB hard disk space for install. At least 1 GB available for data storage and expansion capacity.

Network Install: Network card and connection to drive containing software.

CD-ROM Install: CD-ROM drive

Microsoft Access 2000 installed and operating properly

Cognos Power Play and Cognos Transformer installed and operating properly

Windows 9x platform: DCOM 98 installed

Recommended:

Pentium III or better processor

128 MB RAM

2. Installing the software

Before installing the NaRSDAT software, ensure that both Microsoft Access and Cognos Powerplay Transformer and Cognos PowerPlay are installed. After installing Cognos Transformer, open the program and select File – Preferences. Under the directory tab, set the default directories as follows:

- Model Directory – C:\NaRSDAT\NaRSDAT Models
- Data Source Directory – C:\NaRSDAT
- Power Cube Directory – C:\NaRSDAT\NaRSDAT Cubes

The program will not be able to operate and place the files in the appropriate directories if these defaults are not set in advance. If you miss this step before installation and data

migration, proceed as directed above and change the directories. Then, reload the data cubes from the NaRSDAT application.

Place the NaRSDAT CD in the CDROM drive or connect to the network drive containing the executable. If your CDROM drive has autoplay enabled, the installation will begin automatically. If not, Press "Start" and "Run" and type "D:\setup.exe" where D:\ is the drive letter for your CDROM drive.

If you would like to accept the default install location, Click the "Install" button. You may choose to install to any directory you like by pressing the "Change directory" button. NaRSDAT will then install to the chosen directory.

NaRSDAT will create a group in the Programs menu on the Start bar. To run NaRSDAT, Navigate the Start menu to the NaRSDAT group and select NaRSDAT.

B. GETTING STARTED

When you first install NaRSDAT, the data warehouse and Dimensional cubes will be blank. Start by importing your first data extracts into NaRSDAT. Place the data sources for the legacy systems in the computer. You can import the data from a disk, a CDROM, or a network connection. There is a practice set of data supplied with the NaRSDAT distribution under the NaRSDAT Data directory on the CD.

Open NaRSDAT and press the "Import" button. You will be at a screen that has a button for each of the legacy database extracts. Press the desired button for the import process. The "File Open" dialogue box will appear. Navigate to the location of the file to be imported. Select the file to be imported and select "Open". Depending on the size of the file, this process may take a minute or two for each file. Also, there are two files to imported for TFMMS; one enlisted and one officer. Be sure you import both files before proceeding. Press the "Return" button to continue.

After all the data has been imported, it is time to migrate the data. The migration process moves the data from the legacy database format to the data warehouse format. Press the Build button. The Migrate form will appear to ensure that this is what you want to do. Press the Build button when ready to proceed. This process will take several

minutes to complete. Please be patient. There is a lot of data to format and move. Once this is complete, press the Return Button.

Now you are ready to build the power cubes and populate the cube models. Press the Build Cubes button. This will take you to the Build Cube form. When you are ready, press the Build button again, the cube building begins. Again, this is a lengthy process. Be patient.

Once this is complete, your data warehouse is loaded and ready to run. You may now exit the NaRSDAT application. If you wish to access the data warehouse directly, open Microsoft Access and select the NaRSDATPrototype database in the C:\NaRSDAT folder. You can now view error reports for the import process. This will also allow the you to correct any import errors that are required. You can also run basic custom queries from Microsoft Access. Do not change or delete any existing queries. NaRSDAT will not operate properly if these queries are modified or deleted. If this happens, reinstall the NaRSDAT application.

If you want to continue to the data mining utilities, you can access the predefined reports from the C:\NaRSDAT\NaRSDAT Reports directory. Just double click any of the reports listed. The view of the report can be changed to give you the report that you need. Once the report is formatted the way you want it, select File-Save As and save the new report. The next time you access Cognos, the report will be available. For more information on how to format a report, see the Cognos PowerPlay help files supplied with the default installation.

Each time you upload new legacy data files, you must repeat the entire NaRSDAT import process. It is important to note that all files should be imported together to preclude data from being missed during the import process. If a new member is listed on orders in the Manday database and imported to the Orders table but the Global database has not been imported and updated to the Member table, the orders will not be included in the Orders table.

C. TROUBLESHOOTING

For installation problems, make sure your system meets the minimum system requirements. Make sure all software required by NaRSDAT is installed before running the program. The lack of library references for Microsoft Access or Cognos PowerPlay will cause the program to install and run incorrectly or not at all.

For operational problems, Make sure you have selected the proper files to import to each of the legacy tables. If you import an incorrect file, open Microsoft Access and open the table with the bad data. Click "Edit" on the file menu and click the "Select All" menu item. Next, select "Delete record" on the same menu. Now, return to the NaRSDAT import menu and import the proper file again.

Improper files will not import to any data warehouse tables except the Active Unit and Reserve Unit. These tables will not change very often after the first import. If the data in these two tables become corrupted, remove the existing data by the same procedures as listed above for importing the wrong file. Then migrate the data again. This should correct the corrupt data warehouse entries.

For general application problems, you can search the Microsoft Knowledge Base for any errors reported by Windows, Visual Basic, or Access. Many common problems have solutions already posted in this area. For further information, you can consult several of the newsgroups listed in the Help section of this user's manual.

D. ADDITIONAL HELP

Help for most problems can be found in the Microsoft Knowledge Base. The articles are grouped by Article Ids. You can perform a search from the main Microsoft technical support page at HTTP://www.microsoft.com/technet. Cognos applications also have a support site at HTTP://www.cognos.com. However, there is direct technical support provided under the DoN Cognos contract. Contact the Naval Reserve representative at Cognos directly for the appropriate technical support number.

Additional assistance can be sought in several Newsgroups. The newsgroups that apply directly to this software are as follows:

Microsoft.VisualBasic.

Microsoft.Access.

These news groups can be accessed through most ISPs. Use your default mail program or use a dedicated newsgroup application for accessing the postings. You should first search the groups for articles that apply to the specific problem you are encountering first. Only after a thorough search should you post a new message. If you post a request that has been answered before, many newsgroup users will ignore your question. This is considered very improper in newsgroup circles.

As a last resort, you can inspect the Visual Basic code listed in the next Appendix E of this document. Review the code for specific errors specified during the application's operation. If an error is detected, you can recompile the program and run it again.

Good luck.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX E. VISUAL BASIC AND SQL CODE

A. VISUAL BASIC CODE

```
frmDataWarehouse - 1

      Private Sub cmdBuild_Click()

      'Call subroutine to build reports and cubes
      Call BuildCube
      End Sub

      Private Sub cmdReturn_Click()

      'Reload the Main form
      Unload frmDataWarehouse
      frmMain.Show

      End Sub

frmMain - 1

      Private Sub cmdCube_Click()

      'Hide the current form and show the Import form
      frmMain.Hide
      frmDataWarehouse.Show

      End Sub

      Private Sub cmdExitNaRSDAT_Click()

      'End the program
      End

      End Sub

      Private Sub cmdImportForm_Click()

      'Hide the current form and show the Import form
      frmMain.Hide
      frmNaRSDATImport.Show

      End Sub

      Private Sub cmdMigrateData_Click()

      'Hide the current form and show the Migrate form
      Unload frmMain
      frmMigrate.Show

      End Sub

frmMigrate - 1
```

```
Private Sub cmdMigrateData_Click()

'Call the MigateData subroutine
Call MigrateData

End Sub

Private Sub cmdReturn_Click()

'Reload the Main form
Unload frmMigrate
frmMain.Show

End Sub
```

frmNaRSDATImport - 1

```
'Project:      NaRSDAT prototype for CNRF
'Date:         July 2001
'Programmer:   LCDR Scott A. Langley
'Description:      This prototype is designed to implement
'              the cleaning and importing functions
'              required by NaRSDAT.  It will allow the
'              user to select the import files, note
'              any discrepencies in the process, and exit
'              to the warehouse or OLAP application.
'                  It is designed with a main interface
'              screen and a seperate form and subroutine
'              for each legacy database import.
'                  It will also allow the user to open the
'              data warehouse or data mining application on
'              exit.

Option Explicit

Dim strFileName As String
Dim strFileType As String

Private Sub cmdReturn_Click()

'Return to the main form
Unload frmNaRSDATImport
frmMain.Show

End Sub

Private Sub cmdGlobal_Click()

'Initialize variables
strFileName = ""
strFileType = ""

'Find file path by allowing user to select file
frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
```

```
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "Global"

'Import data
Call FileImport(strFileName, strFileType)

'File specific formating

'Dimension variables for the migration process
Dim AccessApp As Access.Application
Dim strDBPath As String
Dim dbsNow As Database
Dim rstBIN As Recordset
Dim intNewBIN As Long
Dim strNewBIN As String

'Define the database path
strDBPath = App.Path & "\NaRSDATPrototype.mdb"

'Open the NaRSDAT database
Set AccessApp = New Access.Application
AccessApp.OpenCurrentDatabase strDBPath

'Append data for empty BIN data
Screen.MousePointer = 11
AccessApp.DoCmd.OpenQuery ("GlobalNoBINQuery")

Set dbsNow = CurrentDb
Set rstBIN = dbsNow.OpenRecordset("GlobalNoBIN")

'Create new BIN numbers
strNewBIN = ""
rstBIN.MoveFirst
For intNewBIN = 1 To rstBIN.RecordCount
    strNewBIN = "Z" & intNewBIN
    rstBIN.Edit
    rstBIN!BIN = strNewBIN
    rstBIN.Update
    rstBIN.MoveNext
Next

'Update the data to the Global table
AccessApp.DoCmd.OpenQuery ("GlobalBINUpdateQuery")

Screen.MousePointer = 0

End Sub

Private Sub cmdManday_Click()

'Initialize variables
strFileName = ""
strFileType = ""
```

```
'Find file path by allowing user to select file

frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "Manday"

'Import data
Call FileImport(strFileName, strFileType)

'format the data

'Dimension variables for the migration process
Dim AccessApp As Access.Application
Dim strDBPath As String

Screen.MousePointer = 11

'Define the database path
strDBPath = App.Path & "\NaRSDATPrototype.mdb"

'Open the NaRSDAT database
Set AccessApp = New Access.Application
AccessApp.OpenCurrentDatabase strDBPath

'Update the data to the Manday table
AccessApp.DoCmd.OpenQuery ("MandayRUICConversionQuery")

Screen.MousePointer = 0

End Sub

Private Sub cmdRIMSFM_Click()

'Initialize variables
strFileName = ""
strFileType = ""

'Find file path by allowing user to select file

frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "RIMSFM"

'Import data
Call FileImport(strFileName, strFileType)

End Sub
```

```
Private Sub cmdRTSS_Click()

'Initialize variables
strFileName = ""
strFileType = ""

'Find file path by allowing user to select file
frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "RTSS"

'Import data
Call FileImport(strFileName, strFileType)

End Sub

Private Sub cmdTFMMS_Click()

'Initialize variables
strFileName = ""
strFileType = ""

'Find file path by allowing user to select file

frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "TFMMS"

'Import data
Call FileImport(strFileName, strFileType)

End Sub

Private Sub cmdWinPAT_Click()

'Initialize variables
strFileName = ""
strFileType = ""

'Find file path by allowing user to select file
frmCommonDialog.CommonDialog.InitDir = "c:\NaRSDAT"
frmCommonDialog.CommonDialog.ShowOpen
strFileName = frmCommonDialog.CommonDialog.FileName

'set data type
strFileType = "Winpat"

'Import data
Call FileImport(strFileName, strFileType)
```

```
    End Sub
```

Routines - 1

```
    Option Explicit

    Sub FileImport(strFilePath As String, strFileName As String)

    'This subroutine uses the file type and file path to import the
    'required data to the legacy database table in NaRSDAT

    On Error GoTo ErrHandler

    'define subroutine variables
    Dim AccessApp As Access.Application
    Dim strDBPath As String

    'set the path to the database
    strDBPath = App.Path & "\NaRSDATPrototype.mdb"

    'Open the database
    Set AccessApp = New Access.Application
    AccessApp.OpenCurrentDatabase strDBPath

    'Transfer the data to the table
    Screen.MousePointer = 11
    AccessApp.DoCmd.TransferText      acImportDelim,      strFileName,
    strFileName, strFilePath
    Screen.MousePointer = 0

    'return to the previous subroutine
    GoTo ExitSub:

    'error handling routine
    ErrHandler:
        MsgBox "Error: " & Err.Number & vbCrLf & Err.Description

    ExitSub:

    'close out the database
    AccessApp.Quit
    Set AccessApp = Nothing

    End Sub

    Sub MigrateData()

    'This subroutine migrates the legacy data to the data
    'warehouse tables

    'Dimension variables for the migration process
    Dim AccessApp As Access.Application
    Dim strDBPath As String
```

```
'Define the database path
strDBPath = App.Path & "\NaRSDATPrototype.mdb"

'Open the NaRSDAT database
Set AccessApp = New Access.Application
AccessApp.OpenCurrentDatabase strDBPath

'run the data warehouse fill queries
Screen.MousePointer = 11
AccessApp.DoCmd.OpenQuery ("ReserveUnitFillQuery")
AccessApp.DoCmd.OpenQuery ("ActiveUnitFillQuery")
AccessApp.DoCmd.OpenQuery ("MemberFillQuery")
AccessApp.DoCmd.OpenQuery ("BilletFillQuery")
AccessApp.DoCmd.OpenQuery ("MemberInBilletFillQuery")
AccessApp.DoCmd.OpenQuery ("OrdersFillQuery")
AccessApp.DoCmd.OpenQuery ("OrdersFillMoneyQuery")
AccessApp.DoCmd.OpenQuery ("MakeMoneySum")
AccessApp.DoCmd.OpenQuery ("OrdersFillMoneyTableQuery")

'Fix rate information
AccessApp.DoCmd.OpenQuery ("OfficerRateUpdateQuery")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery1")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery2")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery3")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery4")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery5")
AccessApp.DoCmd.OpenQuery ("EnlistedRateUpdateQuery6")
Screen.MousePointer = 0

'close out the database
AccessApp.Quit
Set AccessApp = Nothing

End Sub

Sub BuildCube()

'Dimension the Cognos application
Dim AccessApp As Access.Application
Dim CognosApp As CognosTransformer.Application
Dim TrfModel As Object
Dim strCubePath As String
Dim strStarDBPath As String

'Set the star schema path and data cube model path
strStarDBPath = App.Path & "\NaRSDATStar.mdb"
strCubePath = App.Path & "\NaRSDAT Cube Models\"

'Open the Access star schema database
Set AccessApp = New Access.Application
AccessApp.OpenCurrentDatabase strStarDBPath

'run the data warehouse fill queries
Screen.MousePointer = 11
AccessApp.DoCmd.OpenQuery ("MandayEnlistedStarFillQuery")
```

```
AccessApp.DoCmd.OpenQuery ("MandayEnlistedStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("MandayEnlistedStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("MandayEnlistedStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("MandayOfficerStarFillQuery")
AccessApp.DoCmd.OpenQuery ("MandayOfficerStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("MandayOfficerStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("MandayOfficerStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("RPNEnlistedStarFillQuery")
AccessApp.DoCmd.OpenQuery ("RPNEnlistedStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("RPNEnlistedStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("RPNEnlistedStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("RPNOfficerStarFillQuery")
AccessApp.DoCmd.OpenQuery ("RPNOfficerStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("RPNOfficerStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("RPNOfficerStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("ManpowerEnlistedStarFillQuery")
AccessApp.DoCmd.OpenQuery ("ManpowerEnlistedStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("ManpowerEnlistedStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("ManpowerEnlistedStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("ManpowerEnlistedStarFillQuery5")
AccessApp.DoCmd.OpenQuery ("ManpowerOfficerStarFillQuery")
AccessApp.DoCmd.OpenQuery ("ManpowerOfficerStarFillQuery2")
AccessApp.DoCmd.OpenQuery ("ManpowerOfficerStarFillQuery3")
AccessApp.DoCmd.OpenQuery ("ManpowerOfficerStarFillQuery4")
AccessApp.DoCmd.OpenQuery ("ManpowerOfficerStarFillQuery5")

Screen.MousePointer = 0

'close out the database
AccessApp.Quit
Set AccessApp = Nothing

'Build Manday Enlisted Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "MandayEnlisted.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'Build Manday Officer Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "MandayOfficer.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'Build Manpower Enlisted Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "ManpowerEnlisted.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
```

```
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'Build Manpower Officer Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "ManpowerOfficer.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'Build RPN Enlisted Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "RPNEnlisted.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'Build RPN Officer Cube
Set CognosApp = New CognosTransformer.Application
CognosApp.OpenModel (strCubePath & "RPNOfficer.mdl")
CognosApp.CurrentModel.GenerateCategories
CognosApp.CurrentModel.Update
CognosApp.CurrentModel.CreateMDCFiles
CognosApp.CurrentModel.Close
CognosApp.Quit

'close out the process
Set CognosApp = Nothing

End Sub
```

B. SQL CODE

1. NaRSDAT Prototype.MDB Code

Query: ActiveUnitFillQuery

```
SQL
INSERT INTO ActiveUnit ( AUIC, CL )
SELECT GLOBAL.BAUIC, GLOBAL.ACMD
FROM [GLOBAL];
```

Query: BilletFillQuery

```
SQL
```

```
      INSERT INTO Billet ( BilletIDNumber, NRA, [RUIC(fk)], [AUIC(fk)],
BilletRate, BilletGrade, PrimaryNEC, SecondaryNEC, ReservePgmCode,
VerticalSubCode, HorizSubCode, SexCode, BilletName, UnitName )
      SELECT [GLOBAL].[BIN], [GLOBAL].[NRA], [GLOBAL].[RUIC],
[GLOBAL].[BAUIC], [GLOBAL].[ARATE], [GLOBAL].[BGRADE], [GLOBAL].[PNEC],
[GLOBAL].[SNEC], [GLOBAL].[RPC], [GLOBAL].[V], [GLOBAL].[HZ],
[GLOBAL].[SX], [GLOBAL].[BILTITLE], [GLOBAL].[UNIT]
      FROM [GLOBAL];
```

Query: BilletOfficerRateUpdateQuery

```
      SQL
      UPDATE Billet SET Billet.BilletRate =
Mid([Billet].[BilletRate],1,4)
      WHERE ((([Billet]![BilletGrade]) Like "O[1-9]" Or
([Billet]![BilletGrade]) Like "W[1-4]"));
```

Query: EnlistedRateUpdateQuery1

```
      SQL
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,2)
      WHERE ((([Member]![PGrade]) Like "E[4-7]") AND
(Len([member]![irate])=3));
```

Query: EnlistedRateUpdateQuery2

```
      SQL
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,3)
      WHERE ((([member]![PGrade]) Like "E[4-7]") AND
(Len([member]![irate])=4));
```

Query: EnlistedRateUpdateQuery3

```
      SQL
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,2)
      WHERE ((([member]![pgrade]) Like "E[8-9]") AND
(Len([member]![irate])=4));
```

Query: EnlistedRateUpdateQuery4

```
      SQL
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,3)
      WHERE ((([member]![pgrade]) Like "E[8-9]") AND
(Len([member]![irate])=5));
```

Query: EnlistedRateUpdateQuery5

```
      SQL
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,2)
      WHERE ((([member]![pgrade]) Like "E[1-3]") AND
((Len([member]![irate]))=4));
```

Query: EnlistedRateUpdateQuery6

```
      SQL
```

```
      UPDATE Member SET Member.IRate = Mid([member].[irate],1,3)
      WHERE ((([member]![pgrade]) Like "E[1-3]") AND
((Len([member]![irate]))=5));
```

Query: GlobalBINUpdateQuery

```
      SQL
      UPDATE [GLOBAL] INNER JOIN GlobalNoBIN ON
[GLOBAL].[SSN]=[GlobalNoBIN].[SSN] SET [Global].BIN =
[GlobalNoBIN].[BIN]
      WHERE ((([GLOBAL]![ID])=[GlobalNoBIN]![ID]));
```

Query: GlobalNoBINQuery

```
      SQL
      SELECT [GLOBAL].[ID], [GLOBAL].[NRA], [GLOBAL].[RBSC],
[GLOBAL].[ECPC], [GLOBAL].[CHG], [GLOBAL].[UMUIC], [GLOBAL].[BAUIC],
[GLOBAL].[ARATE], [GLOBAL].[V], [GLOBAL].[HZ], [GLOBAL].[SX],
[GLOBAL].[PNEC], [GLOBAL].[SNEC], [GLOBAL].[BENDT],[GLOBAL].[ADTE],
[GLOBAL].[IRATE], [GLOBAL].[ISX], [GLOBAL].[SSN], [GLOBAL].[MOB],
[GLOBAL].[NAME], [GLOBAL].[IRAD], [GLOBAL].[Q], [GLOBAL].[M],
[GLOBAL].[Y], [GLOBAL].[NECP], [GLOBAL].[NECS], [GLOBAL].[TRUIC],
[GLOBAL].[ABSC], [GLOBAL].[UNIT], [GLOBAL].[RPC], [GLOBAL].[BILTITLE],
[GLOBAL].[OE], [GLOBAL].[REC], [GLOBAL].[GROUPS], [GLOBAL].[PEBD],
[GLOBAL].[DOR], [GLOBAL].[DOB], [GLOBAL].[DTASG], [GLOBAL].[IAP],
[GLOBAL].[IAPDATE], [GLOBAL].[PGRADE], [GLOBAL].[BGRADE],
[GLOBAL].[CMD], [GLOBAL].[PGCD], [GLOBAL].[AUT], [GLOBAL].[ACMD],
[GLOBAL].[APG], [GLOBAL].[AUC], [GLOBAL].[AMGR], [GLOBAL].[SMGR],
[GLOBAL].[PRI], [GLOBAL].[BPRI], [GLOBAL].[A], [GLOBAL].[I],
[GLOBAL].[D], [GLOBAL].[PCMD], [GLOBAL].[PNRA], [GLOBAL].[BIN],
[GLOBAL].[CITY], [GLOBAL].[ST], [GLOBAL].[ZIP], [GLOBAL].[ZIP+4],
[GLOBAL].[RUIC]
      INTO GlobalNoBIN
      FROM [GLOBAL]
      WHERE ((([GLOBAL].[BIN]) Is Null));
```

Query: MakeMoneySum

```
      SQL
      SELECT [OrdersFillMoneyQuery].[SDN],
[OrdersFillMoneyQuery].[SumOfAccPayableSum],[OrdersFillMoneyQuery].[Sum
OfExpendSum], [OrdersFillMoneyQuery].[FirstOfProgramCatCode],
      [OrdersFillMoneyQuery].[FirstOfSSN],[OrdersFillMoneyQuery].[LastO
fNumDays],  [OrdersFillMoneyQuery].[LastOfTravDays],
[OrdersFillMoneyQuery].[LastOfBeginDate]
      INTO MoneySum
      FROM OrdersFillMoneyQuery;
```

Query: MandayRUICConversionQuery

```
      SQL
      UPDATE Manday SET Manday.RUIC = Right([manday].[ruic],5);
```

Query: MemberFillQuery

```
SQL
INSERT INTO Member ( SSN, Name, IRate, PGrade, DateOfBirth,
DateOfRate, PrimaryNEC, SecondaryNEC, OEFlag, SexCode, IAPStatus,
IAPDate, PayEntryBaseDate, City, State, ZipCode )
SELECT [GLOBAL].[SSN], [GLOBAL].[NAME], [GLOBAL].[IRATE],
[GLOBAL].[PGRADE], [GLOBAL].[DOB], [GLOBAL].[DOR], [GLOBAL].[NECP],
[GLOBAL].[NECS], [GLOBAL].[OE], [GLOBAL].[ISX], [GLOBAL].[IAP],
[GLOBAL].[IAPDATE], [GLOBAL].[PEBD], [GLOBAL].[CITY],
[GLOBAL].[ST], [GLOBAL].[ZIP]
FROM [GLOBAL];
```

Query: MemberInBilletFillQuery

```
SQL
INSERT INTO MemberInBillet ( [SSN(fk)], [BilletIDNumber(fk)],
DateAssigned, IRAD, NextTrainingElement, MonthIRADUpdate,
YearIRADUpdate, MobStatus, DrillPayCode )
SELECT [GLOBAL].[SSN], [GLOBAL].[BIN], [GLOBAL].[DTASG],
[GLOBAL].[IRAD], [GLOBAL].[Q], [GLOBAL].[M], [GLOBAL].[Y],
[GLOBAL].[MOB], [GLOBAL].[ECPC]
FROM [GLOBAL];
```

Query: MoneySumAppendQuery

```
SQL
INSERT INTO Orders ( SDN, AccountPayableSum, ExpenditureSum,
PgmCatCode, [SSN(fk)], DutyDays, TravelDays, ReportDate )
SELECT [MoneySum].[SDN], [MoneySum].[SumOfAccPayableSum],
[MoneySum].[SumOfExpendSum], [MoneySum].[FirstOfProgramCatCode],
[MoneySum].[FirstOfSSN], [MoneySum].[LastOfNumDays],
[MoneySum].[LastOfTravDays], [MoneySum].[LastOfBeginDate]
FROM MoneySum INNER JOIN Orders ON
[MoneySum].[SDN]=[Orders].[SDN];
```

Query: OfficerRateUpdateQuery

```
SQL
UPDATE Member SET Member.IRate = Mid([member].[irate],1,4)
WHERE ((([Member]![PGrade]) Like "O[1-9]" Or ([Member]![PGrade])
Like "W[1-4]"));
```

Query: OrdersFillMoneyQuery

```
SQL
SELECT RimsFM.SDN, Sum(RimsFM.AccPayableSum) AS
SumOfAccPayableSum, Sum(RimsFM.ExpendSum) AS
SumOfExpendSum, First(RimsFM.ProgramCatCode) AS
FirstOfProgramCatCode, First(RimsFM.SSN) AS
FirstOfSSN, Last(RimsFM.NumDays) AS LastOfNumDays,
Last(RimsFM.TravDays) AS LastOfTravDays,
Last(RimsFM.BeginDate) AS LastOfBeginDate
FROM RimsFM
GROUP BY RimsFM.SDN;
```

Query: OrdersFillMoneyTableQuery

```
SQL
UPDATE Orders INNER JOIN MoneySum ON
[Orders].[SDN]=[MoneySum].[SDN] SET
Orders.AccountPayableSum = [MoneySum].[SumOfAccPayableSum],
Orders.ExpenditureSum =       [MoneySum].[SumOfExpendSum],
Orders.PgmCatCode = [MoneySum].[FirstOfProgramCatCode]
WHERE ((([Orders]![SDN])=[MoneySum]![SDN] And
([Orders]![SDN])=[MoneySum]![SDN] And
([Orders]![SDN])=[MoneySum]![SDN]));
```

Query: OrdersFillQuery

```
SQL
INSERT INTO Orders ( SDN, ReportDate, DutyDays, TravelDays,
[SSN(fk)], [PgmMgrCode(fk)], [RUIC(fk)], ReportUIC, TravelCode,
BudgetCat, ReservePgmCode, NumberOfMods, RentalCar, Cancelled,
CourseIDNumber )
SELECT [Manday].[SDN], [Manday].[RPTDATE], [Manday].[DUTYDAYS],
[Manday].[TRVLDAYS],[Manday].[SSN], [Manday].[PMC], [Manday].[RUIC],
[Manday].[RPTUIC], [Manday].[TRVLCODE],   [Manday].[BUDCAT],
[Manday].[RPC], [Manday].[NUMODS], [Manday].[RENTALCAR],
[Manday].[CANAX], [Manday].[CIN]
```

Query: ReserveUnitFillQuery

```
SQL
INSERT INTO ReserveUnit ( Ruic, RuicName, Claimant,
ResourceSponsor )
SELECT [GLOBAL].[RUIC], [GLOBAL].[UNIT], [GLOBAL].[ACMD],
[GLOBAL].[RPC]
FROM [GLOBAL];
```

2. NaRSDAT Star.mdb Code

Query: MandayEnlistedStarfillquery

SQL

```
INSERT INTO MandayEnlistedStar ( SDN, BeginDate, DD, TD,
SSN, PCC, RUIC, BC, RPC )
SELECT[Orders].[SDN],[Orders].[ReportDate],[Orders].[DutyDa
ys],  [Orders].[TravelDays],[Orders].[SSN(fk)],  [Orders].[PgmCatCode],
[Orders].[RUIC(fk)],[Orders].[BudgetCat], [Orders].[ReservePgmCode]
FROM Orders;
```

Query: MandayEnlistedStarfillquery2

SQL

```
UPDATE MandayEnlistedStar INNER JOIN Member ON
MandayEnlistedStar.SSN = Member.SSN SET MandayEnlistedStar.Rate =
[Member].[IRate],MandayEnlistedStar.Rank=[Member].[PGrade],
MandayEnlistedStar.OE = [Member].[OEFlag]
WHERE ((([Member]![SSN])=[MandayEnlistedStar]![SSN]));
```

Query: MandayEnlistedStarFillquery3

SQL

```
UPDATE MandayEnlistedStar INNER JOIN ReserveUnit ON
MandayEnlistedStar.RUIC = ReserveUnit.Ruic SET MandayEnlistedStar.RS =
[reserveUnit].[resourcesponsor],MandayEnlistedStar.Cl=
[reserveunit].[claimant];
```

Query: MandayOfficerStarFillQuery

SQL

```
INSERT INTO MandayOfficerStar ( SDN, BeginDate, DD, TD,
SSN, PCC, RUIC, BC, RPC )
SELECT[Orders].[SDN],[Orders].[ReportDate],[Orders].[DutyDa
ys], [Orders].[TravelDays], [Orders].[SSN(fk)], [Orders].[PgmCatCode],
[Orders].[RUIC(fk)],[Orders].[BudgetCat], [Orders].[ReservePgmCode]
FROM Orders;
```

Query: MandayOfficerStarFillQuery2

SQL

```
UPDATE MandayOfficerStar INNER JOIN Member ON
MandayOfficerStar.SSN = Member.SSN SET MandayOfficerStar.Rate =
[Member].[IRate],MandayOfficerStar.OE=[Member].[OEFlag],MandayOfficerSt
ar.Rank = [Member].[PGrade]
WHERE ((([Member]![SSN])=[MandayOfficerStar]![SSN]));
```

Query: MandayOfficerStarFillQuery3

SQL

```
UPDATE MandayOfficerStar INNER JOIN ReserveUnit ON
MandayOfficerStar.RUIC = ReserveUnit.Ruic SET MandayOfficerStar.RS =
[reserveUnit].[resourcesponsor],MandayOfficerStar.Cl=[reserveunit].[cla
imant];
```

Query: ManpowerEnlistedStarFillQuery

SQL

```
INSERT INTO ManpowerEnlistedStar ( BIN, RUIC )
SELECT [Billet].[BilletIDNumber], [Billet].[RUIC(fk)]
FROM Billet;
```

Query: ManpowerEnlistedStarFillQuery2

SQL

```
UPDATE ManpowerEnlistedStar INNER JOIN MemberInBillet ON
ManpowerEnlistedStar.BIN = MemberInBillet.[BilletIDNumber(fk)] SET
ManpowerEnlistedStar!SSN = [MemberInBillet]![SSN(fk)];
```

Query: ManpowerEnlistedStarFillQuery3

SQL

```
UPDATE ManpowerEnlistedStar INNER JOIN Member ON
[ManpowerEnlistedStar].[SSN]=[Member].[SSN]SET
```

```
ManpowerEnlistedStar.Rate = [Member].[Irate], ManpowerEnlistedStar.Rank
= [Member].[Pgrade], ManpowerEnlistedStar.OE = [member].[OEFlag];
```

Query: ManpowerEnlistedStarFillQuery4

SQL

```
UPDATE ManpowerEnlistedStar INNER JOIN ReserveUnit ON
[ManpowerEnlistedStar].[RUIC]=[ReserveUnit].[Ruic]SET
ManpowerEnlistedStar.RS=[ReserveUnit].[ResourceSponsor],
ManpowerEnlistedStar.CL = [ReserveUnit].[Claimant];
```

Query: ManpowerOfficerStarFillQuery

SQL

```
INSERT INTO ManpowerOfficerStar ( BIN, RUIC )
SELECT [Billet].[BilletIDNumber], [Billet].[RUIC(fk)]
FROM Billet;
```

Query: ManpowerOfficerStarFillQuery2

SQL

```
UPDATE ManpowerOfficerStar INNER JOIN MemberInBillet ON
[ManpowerOfficerStar].[BIN]=[MemberInBillet].[BilletIDNumbe
r(fk)] SET ManpowerOfficerStar!SSN = [MemberInBillet]![SSN(fk)];
```

Query: ManpowerOfficerStarFillQuery3

SQL

```
UPDATE Member INNER JOIN ManpowerOfficerStar ON
[Member].[SSN]=[ManpowerOfficerStar].[SSN] SET ManpowerOfficerStar.Rate
= [Member].[Irate], ManpowerOfficerStar.Rank = [Member].[Pgrade],
ManpowerOfficerStar.OE = [member].[OEFlag];
```

Query: ManpowerOfficerStarFillQuery4

SQL

```
UPDATE ReserveUnit INNER JOIN ManpowerOfficerStar ON
[ReserveUnit].[Ruic]=[ManpowerOfficerStar].[RUIC]SET
ManpowerOfficerStar.RS=[ReserveUnit].[ResourceSponsor],
ManpowerOfficerStar.CL = [ReserveUnit].[Claimant];
```

Query: RPNEnlistedStarFillQuery

SQL

```
INSERT INTO RPNEnlistedStar ( SDN, BeginDate, SSN, PCC,
RUIC, BC, RPC, APSum, ExpSum )
SELECT[Orders].[SDN],[Orders].[ReportDate],[Orders].[SSN(fk
)], [Orders].[PgmCatCode], [Orders].[RUIC(fk)], [Orders].[BudgetCat],
[Orders].[ReservePgmCode],[Orders].[AccountPayableSum],[Orders].[Expend
itureSum]
FROM Orders;
```

Query: RPNEnlistedStarFillQuery2

SQL

```
            UPDATE    Member    INNER    JOIN    RPNEnlistedStar    ON
[Member].[SSN]=[RPNEnlistedStar].[SSN]   SET   RPNEnlistedStar.Rate   =
[Member].[IRate],RPNEnlistedStar.Rank=[Member].[PGrade],
RPNEnlistedStar.OE = [Member].[OEFlag]
            WHERE ((([Member]![SSN])=[RPNEnlistedStar]![SSN]));
```

Query: RPNEnlistedStarFillQuery3

SQL

```
            UPDATE   ReserveUnit   INNER   JOIN   RPNEnlistedStar   ON
ReserveUnit.Ruic   =   RPNEnlistedStar.RUIC   SET   RPNEnlistedStar.RS   =
[reserveUnit].[resourcesponsor],RPNEnlistedStar.Cl=[reserveunit].[claim
ant];
```

Query: RPNOfficerStarFillQuery

SQL

```
            INSERT INTO RPNOfficerStar ( SDN, BeginDate, SSN, PCC,
RUIC, BC, RPC, APSum, ExpSum )
            SELECT[Orders].[SDN],[Orders].[ReportDate],[Orders].[SSN(fk
)],  [Orders].[PgmCatCode],  [Orders].[RUIC(fk)],  [Orders].[BudgetCat],
[Orders].[ReservePgmCode],[Orders].[AccountPayableSum],
[Orders].[ExpenditureSum]
            FROM Orders;
```

Query: RPNOfficerStarFillQuery2

SQL

```
            UPDATE Member INNER JOIN RPNOfficerStar ON Member.SSN =
RPNOfficerStar.SSN SET  RPNOfficerStar.Rate     =     [Member].[IRate],
RPNOfficerStar.Rank = [Member].[PGrade], RPNOfficerStar.OE =
            [Member].[OEFlag]
```

Query: RPNOfficerStarFillQuery3

SQL

```
            UPDATE   ReserveUnit   INNER   JOIN   RPNOfficerStar   ON
[ReserveUnit].[Ruic]=[RPNOfficerStar].[RUIC]  SET  RPNOfficerStar.RS  =
[reserveUnit].[resourcesponsor],RPNOfficerStar.Cl=[reserveunit].[claima
nt];
```

LIST OF REFERENCES

1. Department of Defense, *Information Technology Standards Guidance*, Government Printing Office, 1997.

2. Totushek, John B., RADM, USNR, "Naval Reserve Statement." [http://www.navres.navy.mil/navresfor/data/docs/leadingchg/vision.html]. Oct 2000.

3. Kotter, John P., *Leading Change*, Harvard Business School Press, 1996.

4. Totushek, John B., RADM, USNR, "Overview: US Naval Reserve Force." [http://www.navres.navy.mil/navresfor/data/pages/overview.html]. Oct 2000.

5. Brackett, Michael H., *The Data Warehouse Challenge: Taming Data Chaos*, pp. 421, Wiley Computer Publishing, 1996.

6. Humphries, Hawkins, and Dy, *Data Warehousing: Architecture and implementation*, pp. 34, Prentice Hall, 1999.

7. Turban and Aronson, *Decision Support Systems and Intelligent Systems*, Fifth Edition, pp. 126, Prentice Hall, 1998.

THIS PAGE INTENTIONALLY LEFT BLANK

BIBLIOGRAPHY

Kroenke, David M., *Database Processing: Fundamentals, Design & Implementation*, Seventh Edition, Prentice Hall, Upper Saddle River, NJ, 2000.

Brackett, Michael H., *The Data Warehouse Challenge: Taming Data Chaos*, Wiley Computer Publishing, New York, 1996.

Humphries, Hawkins, and Dy, *Data Warehousing: Architecture and implementation*, Prentice Hall, Upper Saddle River, NJ, 1999.

Turban and Aronson, *Decision Support Systems and Intelligent Systems*, Fifth Edition, Prentice Hall, Upper Saddle River, NJ, 1998, pg. 126.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, VA

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, CA

3. Commander, Naval Reserve Force
   Assessment Division, N8
   New Orleans, LA
   drowe@cnrf.nola.navy.mil